



Sistema multiplataforma para aprendizaje de vocabulario contextualizado

Juan Miguel Ruiz Ladrón

Memoria de Trabajo Fin de Grado
Grado en Ingeniería Informática

Curso 2015/2016



ESCUELA SUPERIOR DE INGENIERÍA
INGENIERÍA INFORMÁTICA

Sistema multiplataforma para aprendizaje de vocabulario contextualizado

- Departamento: Ingeniería Informática y Filología Francesa y Inglesa
- Director del proyecto: Manuel Palomo Duarte
- Co-dirección del proyecto: Anke Berns
- Autor del proyecto: Juan Miguel Ruiz Ladrón

Fdo: Juan Miguel Ruiz Ladrón

Agradecimientos

Gracias a mi familia, amigos y a mi pareja por el apoyo en los peores momentos del proyecto.

Gracias a los alumnos colaboradores, Mercedes Paéz Piña, Alicia Garrido Guerrero y Salvador Reyes Sánchez (Departamento de Filología Francesa e Inglesa), de Anke Berns, por su dedicación a la hora de realizar las pruebas y dar un feedback objetivo y profesional.

Éste proyecto no se podría haber llevado a cabo sin la asistencia recibida por parte de Manuel Palomo Duarte y Anke Berns, especialmente por su insistencia. ¡Muchas gracias!

Resumen

Actualmente, el uso de aplicaciones educativas en dispositivos móviles se ha generalizado. Existen infinidad de aplicaciones para mejorar el uso de un idioma en concreto, a menudo con un contenido estático aunque estudiado, pero que no profundiza más allá.

Nuestro sistema, haciendo uso de un anterior juego que se realizaba en papel, consiste en adivinar una palabra, oculta en una frase o preguntada, con la ayuda de una pista. Ésta forma de proceder, en papel, necesita de infinidad de horas para corregir los papeles de un grupo de alumnos. El paso del juego de papel, al de la aplicación móvil ha dado lugar a muchas mejoras, descritas en el documento, además de proporcionar herramientas de corrección y monitorización que, de otra forma, hubiera sido imposible de realizar.

El proyecto consta de tres partes:

- **App del alumno:** Disponible en versión para Android y para un PC con Java instalado. Los alumnos juegan a través de éste cliente y evalúan las definiciones que juegan.
- **Portal web del profesor:** Portal de monitorización y gestión del contenido y usuarios del sistema. En él, el docente tiene total control sobre sus grupos de alumnos, pudiendo tener todos los que estime necesarios en el curso.
- **Servidor web:** Servidor donde montaremos la base de datos y almacenaremos los datos de los usuarios, el contenido y el resultado de sus partidas para su posterior uso en la elaboración de gráficas de monitorización.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2016 Juan Miguel Ruiz Ladrón.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura del documento	2
2. Descripción general	3
2.1. Descripción	3
2.2. Características del sistema	3
2.2.1. Componentes del sistema	3
2.2.2. Colaboradores	4
3. Planificación	5
3.1. Metodología de desarrollo	5
3.2. Fase inicial	6
3.3. Fase de análisis	6
3.4. Fase de aprendizaje	6
3.5. Fase de diseño	7
3.6. Fase de desarrollo	7
3.7. Pruebas y correcciones	7
3.8. Redacción de la memoria	7
3.9. Diagrama de Gantt	8
4. Análisis	9
4.1. Objetivos del proyecto	9
4.2. Especificación de requisitos del sistema	9
4.2.1. Requisitos de interfaces externas	9
4.2.2. Requisitos funcionales	12
4.2.3. Requisitos de rendimiento	12
4.2.4. Restricciones de diseño	13
4.2.5. Requisitos de sistema software	13
4.3. Diagramas de casos de uso	13

4.3.1. Casos de uso del usuario de la app cliente	13
4.3.2. Casos de uso del profesor	23
4.3.3. Casos de uso del administrador	30
4.4. Modelo conceptual de datos	34
4.4.1. Diagrama Entidad-Relación	35
4.5. Modelo de comportamiento	36
4.5.1. Diagramas de secuencia y contrato de las operaciones	36
5. Diseño	63
5.1. Arquitectura	63
5.1.1. Física	63
5.1.2. Lógica	64
5.2. Diseño físico	65
5.3. Diseño de clases	73
5.4. Interfaz gráfica	74
6. Implementación	101
6.1. Layout y resolución	101
6.2. Internacionalización	101
6.3. Multiplataforma	102
6.4. Creación de grupos	102
6.5. Subida de contenido a los grupos	102
6.6. Selección de definiciones	102
6.7. Generador de estadísticas	103
7. Pruebas y validaciones	105
7.1. Pruebas unitarias	105
7.2. Pruebas de integración	106
7.3. Pruebas de jugabilidad	106
7.4. Pruebas de usabilidad	106
8. Conclusiones	107
8.1. Objetivos	107
8.2. Conclusiones personales	107
8.3. Mejoras y ampliaciones	108
A. Apéndice	109
A.1. Lenguaje del cliente	109
A.2. Base de datos	110
A.3. Frontend	110
A.4. Backend	110
A.5. Aseprite, Gimp e Inkscape	111
A.6. Sistema de control de versiones	111
A.7. Documentación del código	111
A.8. Redacción de la memoria	112
A.9. Realización de diagramas: DIA	112
B. Apéndice	113
B.1. Manual de instalación	113
B.1.1. Cliente	113
B.1.2. Interfaz web	113

C. Apéndice	117
C.1. Manual de usuario	117
C.1.1. Manual del cliente	117
C.1.2. Manual de la web	123
Bibliografía	133

Índice de figuras

3.1. Planificación: Diagrama de Gantt	8
4.1. Análisis: Casos de uso del usuario cliente	14
4.2. Análisis: Casos de uso del profesor	24
4.3. Análisis: Casos de uso del administrador	31
4.4. Análisis: Diagrama ER del cliente.	35
4.5. Análisis: Diagrama ER de la web del profesor.	36
4.6. Análisis: Diagrama ER del administrador.	36
4.7. Análisis: Diagrama de flujo del menú principal.	37
4.8. Análisis: Diagrama de flujo del escenario principal.	38
4.9. Análisis: Escenario alternativo al principal del caso Login.	39
4.10. Análisis: Diagrama de flujo de registro en el sistema.	40
4.11. Análisis: Escenario alternativo al caso de uso principal de registro. . .	41
4.12. Análisis: Diagrama de flujo de los créditos del sistema.	41
4.13. Análisis: Diagrama de flujo del bloc de notas	42
4.14. Análisis: Diagrama de flujo de seleccionar grupo	43
4.15. Análisis: Diagrama de flujo de unirse a grupo	44
4.16. Análisis: Diagrama de flujo de enviar solicitudes	45
4.17. Análisis: Casos de uso del profesor	46
4.18. Análisis: Casos de uso del profesor	46
4.19. Análisis: Casos de uso del profesor	47
4.20. Análisis: Diagrama de flujo de añadir definición.	48
4.21. Análisis: Diagrama de flujo de la pantalla de logros.	49
4.22. Análisis: Diagrama de flujo de las estadísticas de usuario.	49
4.23. Análisis: Diagrama de flujo de configurar una ronda.	50
4.24. Análisis: Diagrama de flujo de caso principal de juego	51
4.25. Análisis: Diagrama de flujo de caso alternativo a jugar.	52
4.26. Análisis: Diagrama de flujo de caso alternativo a jugar en caso de error. .	53
4.27. Análisis: Diagrama de flujo de juego cuando se usa pista.	54
4.28. Análisis: Diagrama de flujo juego al guardar palabra en el bloc	55
4.29. Análisis: Diagrama de flujo al reportar una definición	56

4.30. Análisis: Diagrama de flujo de operaciones de gestión de alumnos . . .	57
4.31. Análisis: Diagrama de flujo de validación de alumnos	58
4.32. Análisis: Diagrama de flujo de corrección de definiciones	59
4.33. Análisis: Diagrama de flujo del generador de informes.	60
4.34. Análisis: Diagrama de flujo de la gestión de los profesores.	61
5.1. Diseño: Diagrama de relaciones de la base de datos.	66
5.2. Diseño: Diagrama de interacción entre pantallas	75
5.3. Diseño: Pantalla principal	76
5.4. Diseño: Pantalla de login	77
5.5. Diseño: Pantalla de registro	78
5.6. Diseño: Primera parte de la pantalla de créditos	79
5.7. Diseño: Segunda parte de la pantalla de créditos	80
5.8. Diseño: Pantalla del bloc de notas	81
5.9. Diseño: Pantalla de selección de grupo	82
5.10. Diseño: Pantalla de envío de solicitudes a grupos	83
5.11. Diseño: Pantalla de opciones del grupo	84
5.12. Diseño: Pantalla de configuración de una ronda	85
5.13. Diseño: Pantalla de selección de categorías	86
5.14. Diseño: Pantalla de selección de categorías con algunas seleccionadas	87
5.15. Diseño: Pantalla de menú de estadísticas	88
5.16. Diseño: Primera parte de pantalla de estadísticas	89
5.17. Diseño: Segunda parte de pantalla de estadísticas	90
5.18. Diseño: Tercera parte de pantalla de estadísticas	91
5.19. Diseño: Pantalla de añadir una nueva definición	92
5.20. Diseño: Pantalla de juego	93
5.21. Diseño: Pantalla de número de intentos	94
5.22. Diseño: Pantalla de pista	95
5.23. Diseño: Pantalla de evaluación de la definición	96
5.24. Diseño: Pantalla de motivo de reporte	97
5.25. Diseño: Pantalla de confirmación de palabra guardada en el bloc . . .	98
5.26. Diseño: Pantalla de fin de ronda	99
6.1. Implementación: Generador de estadísticas	103
6.2. Implementación: Definiciones jugadas	104
6.3. Implementación: Definiciones introducidas	104
A.1. Apéndice A: LibGDX logo	109
A.2. Apéndice A: MySQL logo	110
A.3. Apéndice A: CodeIgniter logo	110
A.4. Apéndice A: PHP logo	111
A.5. Apéndice A: Aseprite logo	111
A.6. Apéndice A: Git logo	111
A.7. Apéndice A: L ^A T _E X logo	112
A.8. Apéndice A: Dia logo	112

1.1. Motivación

Al principio, entré en la carrera motivado por el deseo de desarrollar videojuegos en un futuro lo más cercano posible, así que, paralelamente a las clases comencé a informarme sobre las diferentes tecnologías existentes de desarrollo y diseño de videojuegos. A la vez que me documentaba, probaba, creando pequeños juegos y aplicaciones interactivas.

La primera de todas fue un pequeño proyecto, que presenté en el CUSL (Concurso Universitario de Software Libre), llamado Rol & Music, desarrollado en C++ bajo la librería SFML.

La cosa no acabó ahí, me empecé a interesar sobremanera en el framework LibGDX, que usa Java como lenguaje base, y exportable a plataformas móviles.

Un compañero me recomendó a Manuel Palomo para el proyecto, puesto que ya tenía claro que quería realizar un juego y no sabía qué.

Ese año, obtuve prácticas de empresa, bajo la tutoría de Manuel Palomo, para mantener y actualizar un proyecto anterior realizado por un alumno suyo, GuessIt!, desarrollado en Android por Alberto Cejas Sánchez.

Al principio me vino grande, y los problemas de compatibilidad con dispositivos comunes que presentaba al haber actualizado la compilación a un SDK más reciente me hizo plantearme una cosa, cambiar el lenguaje. Fue entonces cuando propuse desarrollar de nuevo GuessIt!, pero en LibGDX, además de incluir muchísimas características nuevas que se comentarán posteriormente.

1.2. Objetivos

El objetivo principal del proyecto es el rediseño y reimplementación de la plataforma de aprendizaje GuessIt! Language Trainer, y la inclusión de nuevas características

a la plataforma, así como el desarrollo de un nuevo portal web de gestión del contenido y la elaboración de un generador de estadísticas interactivo para el personal docente que use la plataforma.

El juego, GuessIt!, es un juego que realiza una evaluación por pares, permitiendo a los usuarios, cumpliendo una serie de requisitos, añadir nuevo contenido jugable a la plataforma.

Para el personal docente, crear un portal web de gestión completa de las aulas o grupos que dé de alta en el sistema. Además de total monitorización de la actividad de sus grupos activos mediante la generación de estadísticas interactivas.

1.3. Estructura del documento

El documento estará compuesto de las siguientes secciones.

- **Introducción:** Breve descripción del proyecto, así como de los objetivos a llevar a cabo y la estructura del propio documento.
- **Descripción general:** Amplia descripción del proyecto, tratando de forma separada el cliente android y la interfaz web.
- **Planificación:** Diagramas de Gantt y fases de la planificación del proyecto.
- **Análisis:** Fase de análisis del sistema, empleando la metodología seleccionada. Se definirán los requisitos funcionales del sistema, diagramas de caso de uso, diagramas de secuencia y contrato de las operaciones.
- **Diseño:** Diseño del sistema, diagramas de secuencia y clases aplicadas al diseño.
- **Implementación:** Aspectos más relevantes durante la implementación del proyecto. Y problemas que han aparecido durante el desarrollo de este.
- **Pruebas y validaciones:** Pruebas realizadas al proyecto.
- **Conclusiones:** Conclusiones obtenidas en la finalización del proyecto.
- **Apéndices:**
 - **Herramientas utilizadas:** Breve explicación de las herramientas usadas en la elaboración del proyecto.
 - **Manual de instalación:** Manual para la instalación del sistema.
 - **Manual de usuario:** Manual de usuario para el correcto uso de la aplicación y de la web.
- **Bibliografía:** Libros y referencias consultadas durante el desarrollo del proyecto.
- **Licencia GNU GFDL:** Texto completo sobre la licencia GNU GFDL en inglés.

Descripción general

2.1. Descripción

El juego es una adaptación a tecnologías móviles de su versión en papel y consistía en adivinar una palabra oculta en una frase o responder a una pregunta sobre una palabra concreta.

Con la adaptación, los alumnos pueden proponer nuevas definiciones y preguntas, lo que antes se tardaría muchísimo en procesar y corregir por el docente.

Los alumnos disponen de una versión Android y una en Java para realizar los ejercicios, conectados a un servidor remoto que les provee de contenido para jugar.

Como novedad, los alumnos no se verán mezclados con gente externa a la aplicación debido a la mecánica de grupos, explicada más adelante.

2.2. Características del sistema

2.2.1. Componentes del sistema

En ésta sección se hará una breve descripción de los componentes del sistema.

Definiciones

Cadenas de texto que son usadas o bien como preguntas, o bien como frase para crear un contexto y así adivinar la palabra oculta.

Categorías

Categorías que dividen las diferentes definiciones según su contexto y el uso de la palabra oculta.

Niveles

Dificultad de las definiciones y palabras a adivinar. Los niveles van del uno al cuatro, cubriendo el nivel A1 del Marco Común Europeo de Referencia para las lenguas (MCER).

Usuarios

Divididos en jugadores, docentes y el administrador del sistema. Cada uno con su cuenta de uso personal, registrada en la base de datos, y con un rol determinado.

Estadísticas

Haciendo uso de una serie de índices almacenados de cada partida, se establecen unos cálculos estadísticos de rendimiento, útiles tanto para docentes como para jugadores.

Bloc de notas

Apartado del cliente, almacenado de forma local, que permite a los jugadores guardar palabras jugadas y agregarles notas personales para su estudio posterior.

Grupos

Organización de los usuarios para su posterior supervisión, un docente puede tener asignado varios grupos de jugadores y/o alumnos. De igual forma, un mismo jugador puede jugar en cualquier cantidad de grupos a los que éste haya sido validado.

2.2.2. Colaboradores

El proyecto ha sido desarrollado de forma individual, tanto la implementación como el diseño, contando con la colaboración de alumnos desempeñando el rol de beta testers.

El paquete de idioma que incluye, por el momento, el alemán, ha sido realizado por un experto.

Los créditos que se encuentran en la aplicación son los siguientes:

Diseño y concepto Manuel Palomo Duarte, Anke Berns, Juan Miguel Ruiz Ladrón

Programación Juan Miguel Ruiz Ladrón

Diseño Gráfico

- **Icono app:** Reutilizado de la versión anterior, gracias a José Carlos Cejas Sánchez.
- **Interfaz web:** Bootstrap 3
- **Interfaz app:** Juan Miguel Ruiz Ladrón

Paquete de idioma:

- **Alemán:** Anke Berns, Alicia Garrido Guerrero, Mercedes Páez Piña, Salvador Reyes Sánchez

3.1. Metodología de desarrollo

Se ha escogido, para la elaboración de éste proyecto, una metodología iterativa e incremental que proporciona, a cada siguiente iteración, nuevas funcionalidades al sistema.

Los principios básicos de ésta metodología son:

- Se puede gestionar las expectativas del cliente de manera regular, puede tomar decisiones en cada iteración. Principalmente porque el cliente a priori **no sabe lo que necesita** y lo va sabiendo a medida que avanza el proyecto.
- El cliente puede comenzar el proyecto con requisitos de alto nivel, quizás no del todo completos, de manera que se vayan refinando en sucesivas iteraciones.
- El cliente puede obtener resultados importantes y usables ya desde las primeras iteraciones.
- Permite conocer el progreso real del proyecto desde las primeras iteraciones y extrapolar si su finalización es viable en la fecha prevista.
- Permite gestionar la complejidad del proyecto.

Por el contrario tiene algunas restricciones a tener en cuenta:

- La disponibilidad del cliente debe ser alta durante todo el proyecto dado que participa de manera continua.
- Cada iteración debe dar como resultado requisitos terminados, de manera que el resultado sea realmente útil para el cliente y no deje tareas pendientes para futuras iteraciones o para la finalización del proyecto.
- Cada iteración ha de aportar un valor al cliente, entregar unos resultados cerrados que sean susceptibles de ser utilizados por él.

Sin embargo es una buena metodología para el desarrollo individual como el que supone éste proyecto.

Ésta metodología no se lleva a cabo de manera estricta, adaptamos sus reglas a las necesidades en el proceso de desarrollo y pruebas.

3.2. Fase inicial

Ésta fase consistió en plantear la idea principal del proyecto, aconsejado por Manuel Palomo Duarte y Anke Berns. Se decidió realizar éste proyecto por las motivaciones explicadas con anterioridad.

3.3. Fase de análisis

Esta etapa está dividida principalmente en las dos partes siguiente:

- **Especificación de los requisitos:** estudio de los requisitos que deberá cumplir el sistema.
- **Recursos necesarios:** recursos necesarios que deberemos usar durante el desarrollo del proyecto.

3.4. Fase de aprendizaje

Ésta fase consistió en aprender todas aquellas características de LibGDX que harían posible la realización del proyecto. Previamente había realizado pequeñas pruebas con éste framework, pero nunca había realizado un proyecto como tal.

Se puede dividir ésta fase en otras más pequeñas como:

- **Aprender Scene2D:** Dentro del framework existen una serie de clases, recogidas en el paquete Scene2D y Scene2D.UI, pensadas para aplicaciones que hagan más uso de una interfaz que de cálculos físicos y animaciones.
- **Aprender HTTP:** Otra de las características del framework es que permite (al estar escrito en Java) realizar peticiones HTTP a servidores remotos, e incluso crear conexiones a través de Socket para el desarrollo de juegos online.
- **Aprender sobre diseño:** No todo es programar, tuve que aprender a usar varios programas, como Gimp, Aseprite e Inkscape para la realización de iconografía y diseño de botones y cuadros de texto para la interfaz.
- **Recordar SQL:** Haciendo uso de los apuntes sobre bases de datos diseñar las consultas necesarias para la realización del proyecto.
- **Desarrollo web:** La web está realizada en CodeIgniter, visto con anterioridad en la asignatura de Programación Web, y al ser fácil de instalar y de manejar, haciendo uso de PHP, ha sido la mejor elección para el desarrollo de la web de gestión del sistema. Además de usar para el diseño Bootstrap 3 y para las gráficas Chart.js.

3.5. Fase de diseño

Antes de comenzar el desarrollo, es necesario diseñar nuestro sistema. Podemos destacar las siguientes partes:

- **Estructura de la base de datos:** Es necesario diseñar una estructura que permita almacenar tanto la información de los usuarios del sistema como de la información con la que trabajaremos.
- **Estructura del cliente:** La distribución de pantallas, la interacción entre éstas y las conexiones que tendrán con la base de datos.
- **Estructura de la web:** Diseño visual y de las herramientas de la interfaz web de monitorización y gestión.

3.6. Fase de desarrollo

Tras la realización del análisis, el diseño y aprender lo necesario, se comienza con el desarrollo, dividido en las siguientes partes:

- **Base de datos:** Necesario empezar por aquí para estructurar la información a almacenar y procesar.
- **Cliente:** Para verificar que las estructuras de información funcionan, se realizó el cliente, tanto para Android como para Java, y realizar primero pruebas locales de conexión.
- **Servidor remoto:** Luego, se preparó un servidor externo para las pruebas de conexión remota y las respuestas de las consultas a la base de datos.
- **Interfaz web:** Por último se realizó la web de gestión del contenido y administración de los usuarios y grupos. Además de realizar una web de administrador para la gestión de elementos clave del sistema (gestión de idiomas, validación de docentes en la plataforma, etc).

3.7. Pruebas y correcciones

Fase esencial del proyecto, se realizó paralelamente al desarrollo. Alumnos colaboradores del departamento de Anke Berns, citados anteriormente, realizaron un feedback semanal para corregir errores y mejorar la sensación del juego, de cara al lanzamiento en un entorno con usuarios reales.

3.8. Redacción de la memoria

La redacción de la memoria se ha realizado conforme se avanzaba en el proyecto, salvo algunas secciones que se tuvieron que desarrollar luego, además de correcciones por cambios en el producto final.

3.9. Diagrama de Gantt

Se muestra a continuación el Diagrama Inicial de Gantt elaborado para la planificación del proyecto.

No es el diagrama definitivo ya que ha estado sujeto a cambios a lo largo del proyecto.

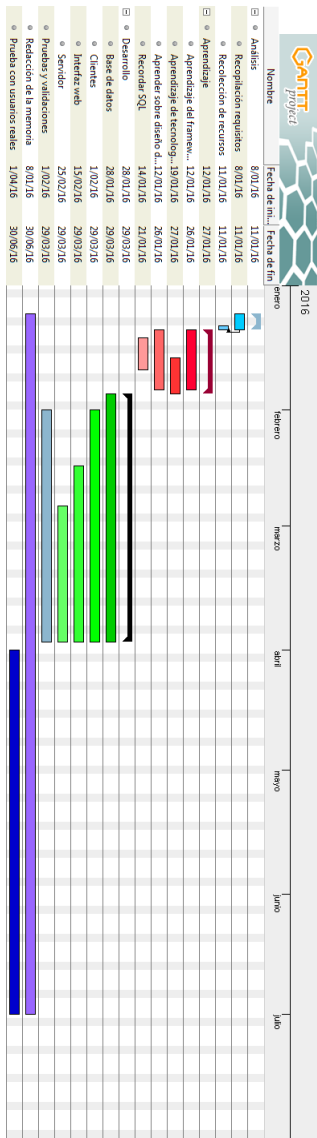


Figura 3.1: Planificación: Diagrama de Gantt

4.1. Objetivos del proyecto

Listaremos los objetivos de éste proyecto:

- **Persistencia de los datos:** Toda la información generada por las partidas jugadas ha de ser almacenada para su posterior procesamiento y monitorización.
- **Gestión del contenido:** En todo momento, el personal docente debe poder gestionar las definiciones a jugar por los usuarios.
- **Gestión de grupos:** Cada docente que use la plataforma debe tener a su disposición herramientas para la creación de grupos donde organizar las distintas comunidades de las que disponga.
- **Monitorización del progreso:** El sistema debe proveer herramientas suficientes para monitorizar el rendimiento de los usuarios del sistema, fragmentado en grupos.
- **Sistema multiplataforma:** El sistema debe desarrollarse para diversas plataformas, siendo los dispositivos móviles los más usados para éste tipo de software.

4.2. Especificación de requisitos del sistema

Para la creación de cualquier producto software, es necesario establecer las distintas condiciones y necesidades que ha de satisfacer. Seguiremos un esquema que nos permita describir los requisitos de una forma metódica y racional.

4.2.1. Requisitos de interfaces externas

Se describirá los requisitos de conexión del software y el hardware, además de la interfaz de usuario.

De la conexión entre software y hardware se encarga el framework y el SDK de Android, necesario para el desarrollo en LibGDX.

Pasamos ahora a definir la interfaz del cliente móvil. Las resoluciones se ajustan automáticamente, siendo la diferencia una proporción respecto a la resolución de la pantalla del dispositivo que lo ejecute.

Respecto a la web, se ajusta a la resolución de pantalla, además de tener un diseño responsivo, facilitando su uso a través de dispositivos móviles.

El cliente móvil, a menos que se indique lo contrario, se manejará a través de la pantalla táctil sólo pulsando en los botones contenidos en la interfaz o en un teclado del propio dispositivo.

Las pantallas que encontraremos en el cliente móvil son las siguientes:

- **Menú principal:** Pantalla inicial de la aplicación donde se muestran las opciones posibles, de login, de registro, créditos y bloc de notas. Las opciones son:
 - Login
 - Registro
 - Créditos
 - Bloc de notas
- **Login:** Pantalla donde introducimos las credenciales del usuario para acceder al juego.
- **Registro:** Pantalla donde rellenamos los datos que pide el juego para registrar un nuevo usuario.
- **Créditos:** Pantalla con la información de los participantes del proyecto.
- **Bloc de notas:** Pantalla donde revisaremos las palabras guardadas y añadimos notas a éstas para su posterior estudio.
- **Seleccionar grupo:** Pantalla donde se nos muestran los grupos a los que tenemos acceso para entrar a jugar.
- **Unirse a grupos:** Pantalla donde se muestran los grupos disponibles en el sistema. Cada grupo se muestra como un botón para enviar una solicitud de acceso al grupo, pendiente de validar.
- **Menú del grupo:** Pantalla con las opciones que podemos seleccionar una vez estemos dentro de un grupo. Las opciones son:
 - Jugar
 - Estadísticas
 - Añadir palabra
- **Jugar:** Pantalla donde configuraremos una ronda de juego.
- **Estadísticas:** Pantalla de menú que muestra las siguientes opciones:
 - Estadísticas
 - Logros

- **Añadir palabra:** Pantalla que se desbloquea, en este caso, a las 50 palabras acertadas dentro del grupo, que sirve para añadir una definición nueva a una palabra existente en el grupo.
- **Estadísticas:** Estadísticas de juego en el grupo, definiciones que el usuario ha propuesto y han sido reportadas y, en general, el rendimiento del usuario en dicho grupo.
- **Logros:** Pantalla donde se muestra el progreso de los diferentes logros dentro del grupo.
- **Juego:** Pantalla donde se nos muestra una definición con la palabra a adivinar oculta, un cuadro de texto donde el usuario propone una palabra, un botón para adivinar y un botón de pista. Si el usuario falla en su intento de adivinar, se muestra una pequeña ventana indicando el número de intentos restantes. Si el usuario falla al menos una vez, se desbloquea el uso de la pista, que muestra una pequeña ventana con la pista relacionada con la palabra. Cuando se falla 3 veces, saltamos a la ventana de evaluación directamente.
- **Evaluar la definición:** Pantalla donde se muestra la definición con la palabra, en verde si se ha acertado dentro de los 3 intentos y en rojo si no, una pregunta de respuesta Sí/No que nos marcará la evaluación de la definición, un botón para reportar, uno para guardar la palabra jugada en el bloc de notas y un botón para enviar la puntuación y pasar a la siguiente palabra.
- **Ventana de reporte:** Pantalla donde muestra, en botones, los motivos para reportar una definición o palabra.
- **Final de ronda:** Pantalla que emerge al jugar 10 definiciones, acertadas o no, que muestra cuantas hemos acertado en la ronda y dos botones, uno para volver a la configuración de la ronda y otro para volver al menú principal del grupo.
- **Interfaz web del docente:** El docente dispone de las siguientes opciones, explicadas brevemente:
 - Crear y administrar sus grupos.
 - Generar estadísticas completas de sus grupos, a través de un formulario especial.
 - Gestionar a los usuarios de sus diferentes grupos, así como validarles el acceso a los mismos.
 - Gestionar el contenido de los grupos, definiciones y palabras.
 - Modificar sus propios datos de contacto y logueo.
- **Interfaz web del administrador:** El administrador del sistema dispone de otras opciones:
 - Gestionar los idiomas disponibles en la plataforma.
 - Gestionar a los docentes y validarles el acceso a la plataforma.
 - Generar estadísticas generales de cualquier grupo del sistema, con motivo de investigación.
 - Modificar sus propios datos de contacto y logueo (no recomendado).

4.2.2. Requisitos funcionales

Los requisitos funcionales a cumplir son los siguientes:

- Salir de la aplicación en cualquier pantalla.
- Login
- Registro
- Seleccionar categoría o categorías a jugar.
- Seleccionar el nivel a jugar.
- Adivinar la / las palabras.
- Puntuar la definición
- Añadir palabra al bloc de notas
- Unirse a grupo
- Entrar en un grupo
- Consultar el bloc de notas
- Recordar credenciales para un acceso rápido
- Ver los créditos
- Generar datos estadísticos de evaluación
- Gestionar la información de los usuarios
- Gestionar el contenido de los grupos
- Gestionar grupos
- Gestionar el idioma
- Gestionar a los docentes
- Generar datos estadísticos de investigación

4.2.3. Requisitos de rendimiento

El rendimiento de la aplicación debe permitir un desempleo agradable de la misma, haciendo mínimas las llamadas al servidor y, las que se necesiten hacer, para realizar operaciones de bajo coste, disminuyendo lo máximo posible el tiempo de respuesta del servidor.

Además de almacenar en caché toda la información posible para disminuir las llamadas al servidor.

Por último, todos aquellos recursos no necesarios, descargarlos de la memoria para economizar el uso de la batería.

4.2.4. Restricciones de diseño

Como comento los puntos del apartado anterior el tiempo de respuesta y minimizar la utilización de recursos tiene que primar sobre cualquier factor. Esta será la principal restricción de diseño que tendrá nuestra aplicación.

Será de valor añadido que el paquete .APK final tenga el menor tamaño posible para abarcar aquellos teléfonos con menos memoria secundaria.

Los videojuegos están pensados como aplicación principal, de forma que no tenga que compartir recursos con otros procesos, por lo que se permitirá que consuma gran parte de los recursos del sistema. Sin embargo, debe tener en cuenta que los dispositivos móviles Android (Java) pueden perder el contexto de la aplicación que ejecutan por lo que el contenido debe ser recuperable.

4.2.5. Requisitos de sistema software

La aplicación deberá cumplir los siguiente requisitos del sistema:

- Deberá funcionar como mínimo para versiones de Android superiores a 2.3.6 dando soporte así a más del 88 % del mercado actual según Google el 9 de Septiembre de 2014.
- El código debe ser mantenible y fácilmente ampliable para futuras versiones.

4.3. Diagramas de casos de uso

Para describir los distintos comportamientos que tendrá el sistema, usaremos el lenguaje de modelado de sistemas *UML*; que representa los requisitos funcionales del sistema, centrandose en que hace y no cómo lo hace.

En primer lugar mostramos el modelo de casos de uso, tanto de la aplicación Android como de la interfaz web, que representa la funcionalidad completa del sistema.

Se ha usado el siguiente esquema:

1. Identificar los usuarios del sistema y los roles que pueden tener.
2. Para cada rol, identificar las distintas formas de interactuar en el sistema. En el caso de nuestro sistema existen tres roles de acceso distintos. Sin embargo, en el cliente Android siempre se logeará con funcionalidades de estudiante mientras que en la interfaz web como profesor, además del administrador.
3. Creación de los casos de uso para todos los objetivos que queramos cumplir.
4. Estructurar dichos casos de uso.

4.3.1. Casos de uso del usuario de la app cliente

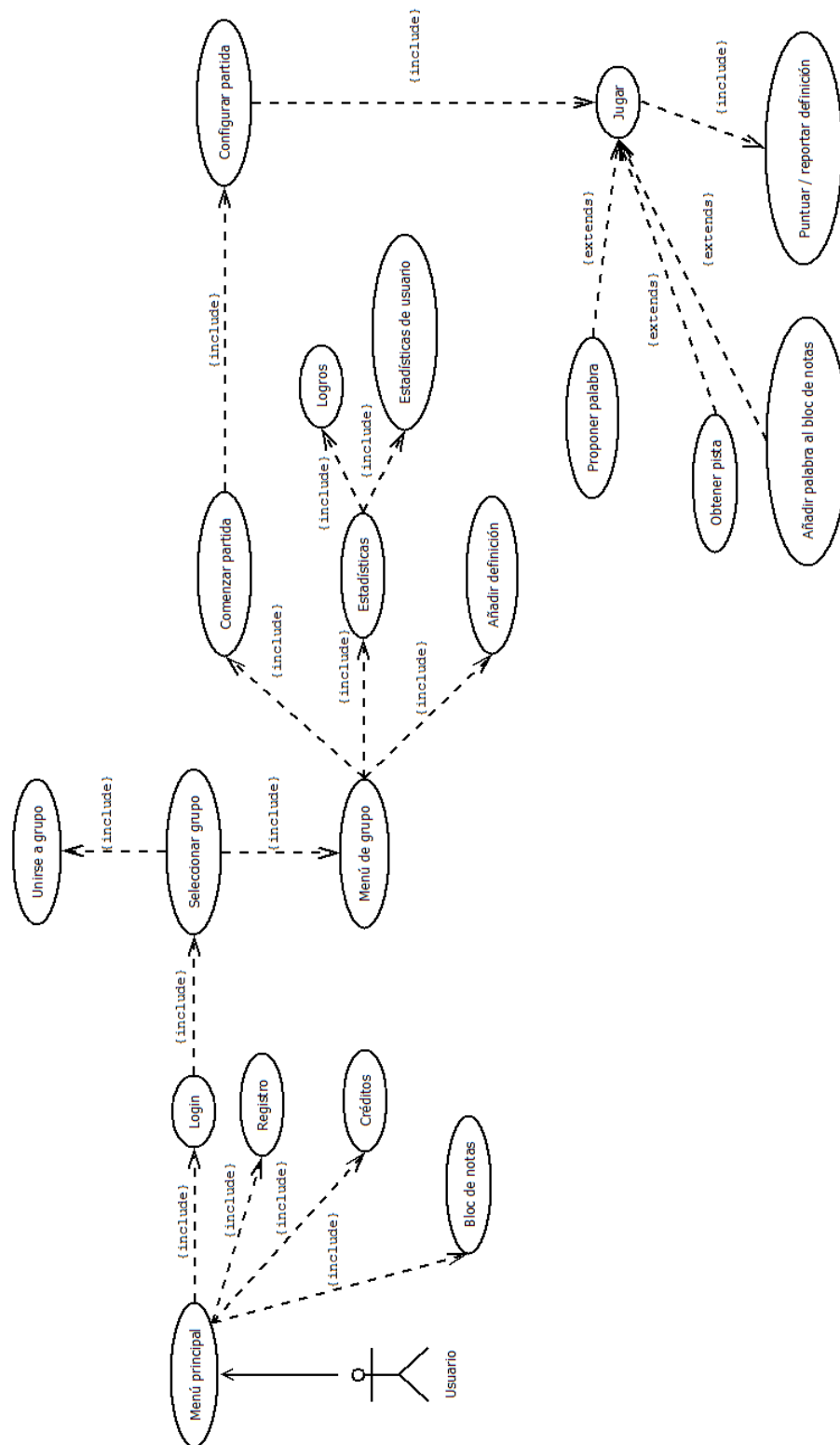


Figura 4.1: Análisis: Casos de uso del usuario cliente

Descripción de los casos de uso

A continuación describimos los casos de uso del usuario del cliente de nuestro sistema. Usaremos una plantilla y una descripción legible y entendible por un usuario no experto.

Caso de uso: Menú principal**Caso de uso** Menú principal

Descripción: El usuario realiza una de las operaciones mostradas en el menú principal de la aplicación.

Actores: Usuario

Precondiciones: Ninguna

Postcondiciones: El usuario entra en la sección correspondiente al botón que ha pulsado.

Escenario principal:

1. El sistema muestra por pantalla el menú inicial del juego.
2. El usuario selecciona una opción de entre las posibles.
3. El sistema carga la *Screen* correspondiente.

Caso de uso: Login**Caso de uso** Login

Descripción: El usuario se loguea en el sistema a través del formulario dado por el mismo.

Actores: Usuario

Precondiciones: El usuario está registrado en el sistema.

Postcondiciones: Usuario logueado en el sistema.

Escenario principal:

1. El sistema muestra la pantalla con los campos de email y password.
2. El usuario introduce sus datos en los campos correspondientes.
3. El sistema “recuerda” la información en los campos de email y password para posteriores usos.
4. El sistema valida los datos del usuario con los almacenados en la base de datos.

5. El sistema muestra la pantalla de selección de grupo.

Extensiones – Flujo alternativo

4a El sistema lanza un error por usuario no registrado en el sistema o datos incorrectos.

4b El usuario debe retroceder al menú principal y registrarse o introducir de nuevo sus datos.

Caso de uso: Registro**Caso de uso Registro**

Descripción: Formulario de registro en el sistema.

Actores: Usuario

Precondiciones: Usuario no registrado en el sistema.

Postcondiciones: Usuario registrado en el sistema.

Escenario principal:

1. El sistema muestra en pantalla los campos a rellenar por el usuario para el registro.
2. El usuario introduce los datos pedidos.
3. El sistema comprueba que los datos del usuario no existen en la base de datos.
4. El usuario queda registrado en el sistema.

Extensiones – Flujo alternativo

3a El usuario existe en la base de datos.

1. El sistema muestra un mensaje de error indicando que el usuario ya existe en el sistema.

3b El usuario introduce datos incorrectos en los campos de registro.

1. El sistema muestra un mensaje de error indicando el campo erróneo.

Caso de uso: Créditos**Caso de uso Créditos**

Descripción: Ver los créditos de los colaboradores y desarrolladores del proyecto.

Actores: Usuario

Precondiciones: Ninguna

Postcondiciones: Los créditos se muestran en pantalla.

Escenario principal:

1. El sistema muestra los datos de las personas implicadas en el proyecto.

Caso de uso: Bloc de notas

Caso de uso Bloc de notas

Descripción: El usuario revisa las palabras guardadas y puede añadir apuntes para mejorar su aprendizaje.

Actores: Usuario

Precondiciones: El usuario debe de haber guardado al menos una palabra jugada.

Postcondiciones: Las palabras son enlazadas a los apuntes que el usuario estime oportunos.

Escenario principal:

1. El sistema muestra un listado con las palabras guardadas por el usuario.
2. El sistema muestra, debajo de cada palabra, un cuadro de texto libre para apuntar cualquier cosa en cualquier idioma.
3. El usuario introduce información que le será útil para mejorar su aprendizaje de la palabra.
4. El sistema guarda los apuntes enlazándolos a la palabra correspondiente.

Caso de uso: Seleccionar grupo

Caso de uso Seleccionar grupo

Descripción: El usuario selecciona el grupo en el que va a jugar.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema y validada su inscripción en los grupos que solicitó.

Postcondiciones: El usuario accede al menú interno del grupo e interactuar con él.

Escenario principal:

1. El sistema muestra un listado con todos los grupos al que el usuario ha sido validado previamente en sus solicitudes de admisión.

2. El usuario pulsa sobre el grupo al que desea jugar.
3. El sistema cambia a la pantalla del menú del grupo.

Extensiones – Flujo alternativo

- 1a El usuario no ha sido validado en ningún grupo por ningún profesor o tutor.

Caso de uso: Unirse a grupo**Caso de uso** Unirse a grupo

Descripción: Lista de grupos donde el usuario envía una solicitud a cada grupo que pulse para unirse.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema.

Postcondiciones: Solicitud de ingreso a los grupos seleccionados.

Escenario principal:

1. El sistema muestra un listado de todos los grupos disponibles en el sistema.
2. El usuario pulsa sobre cada grupo al que desee unirse.
3. El sistema oculta el grupo pulsado de la lista para evitar repeticiones de solicitudes.
4. El sistema envía por petición HTTP una solicitud de ingreso al grupo.

Caso de uso: Menú de grupo**Caso de uso** Menú de grupo

Descripción: Se muestra el menú de grupo con las diferentes secciones, juego, estadísticas, añadir definición.

Actores: Usuario

Precondiciones: El usuario está logueado en el sistema y validado en el grupo.

Postcondiciones: El sistema cambia la pantalla a la opción seleccionada.

Escenario principal:

1. El sistema muestra las opciones disponibles al entrar a un grupo (Jugar, Estadísticas, etc).

2. El usuario pulsa sobre la opción deseada.
3. El sistema cambia a la pantalla correspondiente.

Extensiones – Flujo alternativo

1a El sistema muestra la opción “Añadir definición” si el jugador cumple con los requisitos establecidos.

Caso de uso: Configurar partida

Caso de uso Configurar partida

Descripción: Menú en el que se configura una ronda de juego.

Actores: Usuario

Precondiciones: Usuario logueado, validado en el grupo seleccionado.

Postcondiciones: La ronda comienza con la configuración seleccionada.

Escenario principal:

1. El sistema muestra las opciones de la ronda de juego.
2. El usuario selecciona el nivel a jugar.
3. El usuario pulsa sobre el botón “Seleccionar categorías”.
4. El sistema cambia a la ventana para seleccionar las categorías a jugar.
5. El usuario pulsa sobre cada categoría que desea jugar.
6. El sistema cambia el icono de la categoría seleccionada para observar la selección.
7. El usuario vuelve atrás para continuar con el menú de configuración.
8. El usuario pulsa sobre el botón de jugar.
9. El sistema comienza la ronda de juego.

Extensiones – Flujo alternativo

5a El usuario no selecciona ninguna categoría.

- 1.** El usuario pulsa sobre comenzar la partida.
- 2.** El sistema no comienza la ronda hasta que no se haya seleccionado al menos una categoría.

Caso de uso: Estadísticas

Caso de uso Estadísticas

Descripción: Menú para seleccionar qué ver en éste apartado.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema y validado en el grupo seleccionado.

Postcondiciones: Cambio a la ventana de información estadística correspondiente.

Escenario principal:

1. El sistema muestra el menú de estadísticas.
2. El usuario selecciona entre “Logros” y “Estadísticas generales”.
3. El sistema cambia a la ventana correspondiente.

Caso de uso Logros

Descripción: Ventana de visualización de logros completados.

Actores: Usuario

Precondiciones: Usuario logueado y validado en el grupo.

Postcondiciones: Ninguna

Escenario principal:

1. El sistema muestra la ventana de logros.
2. El usuario puede comprobar su progreso, medido en logros, en ese grupo.

Caso de uso Estadísticas de usuario

Descripción: Ventana en la que se muestra el progreso del usuario de forma detallada.

Actores: Usuario

Precondiciones: Usuario logueado y validado en el grupo.

Postcondiciones: Ninguna

Escenario principal:

1. El sistema muestra la ventana con las estadísticas del usuario.
2. El usuario puede comprobar su progreso de forma detallada.

Caso de uso: Añadir definición**Caso de uso** Añadir definición

Descripción: Formulario que el usuario puede usar para introducir una nueva definición de una palabra en el sistema.

Actores: Usuario

Precondiciones: Usuario logueado y validado en el grupo. Además de haber alcanzado el mínimo para desbloquear ésta característica.

Postcondiciones: Nueva definición, de una palabra existente, en el sistema.

Escenario principal:

1. El sistema muestra el formulario con la palabra a definir, su artículo, nivel y categoría.
2. El usuario introduce los datos.
3. El usuario pulsa sobre el botón de “Enviar”.
4. El sistema envía por petición http el contenido del formulario.
5. El sistema registra la nueva definición en la base de datos.
6. El sistema muestra la ventana “Definición añadida” en el cliente.

Caso de uso: Jugar**Caso de uso** Proponer palabra

Descripción: El usuario propone una solución a la definición dada.

Actores: Usuario

Precondiciones: Usuario logueado, validado en el grupo y ronda de juego configurada.

Postcondiciones:

Escenario principal:

1. El sistema muestra la ventana de juego con el cuadro de texto correspondiente a la palabra a proponer.
2. El usuario escribe la palabra que cree correcta en el cuadro de texto.
3. El usuario pulsa sobre el botón “Adivinar”.

4. El sistema valida la respuesta y muestra la ventana de “Puntuar / Reportar definición”.

Extensiones – Flujo alternativo

- * El usuario pulsa sobre el botón “Pista”
 1. El sistema, si el usuario ha fallado al menos una vez, muestra en una ventana la pista ofrecida en la definición.
 2. El usuario cierra la ventana de pista.
- 3a La palabra introducida no es correcta.
 1. El sistema muestra una ventana de error.
 2. El sistema resta uno al número de intentos que tiene el usuario para adivinar la palabra.
 3. Salto al punto 2.

Caso de uso: Puntuar / Reportar palabra**Caso de uso** Puntuar / Reportar palabra

Descripción: Ventana en la que el usuario ha de responder a una pregunta aleatoria tomando como referencia la definición.

Actores: Usuario

Precondiciones: Usuario logueado, validado en el grupo que ha jugado al menos una definición.

Postcondiciones: Puntuación o reporte almacenado en la base de datos.

Escenario principal:

1. El sistema muestra la ventana de puntuación de la definición.
2. El sistema muestra, además, los botones “Reportar” y “Guardar palabra”.
3. El sistema escoge, al azar, una pregunta de una batería preestablecida y la muestra en pantalla con las opciones “Sí” y “No”.
4. El usuario responde a la pregunta y pulsa sobre “Siguiendo definición”.
5. El sistema envía por http el resultado que se almacena en la base de datos.
6. El sistema vuelve a “Jugar”.

Extensiones – Flujo alternativo

- * El jugador puede guardar la palabra.
 1. El jugador pulsa sobre el botón “Guardar”.
 2. El sistema guarda la palabra de forma local en el dispositivo.
 3. El sistema muestra un mensaje de “Guardado satisfactoriamente”.

4. El usuario vuelve a la ventana de puntuación.
- 4a El usuario reporta la definición.
 1. El usuario pulsa sobre el botón “Reportar”.
 2. El sistema muestra una ventana con los posibles motivos del reporte.
 3. El usuario selecciona uno de esos motivos y cierra la ventana.
 4. El usuario pulsa sobre “Siguiendo definición”.
 5. El sistema envía por http el reporte con el motivo.
 6. El sistema vuelve a “Jugar”.

4.3.2. Casos de uso del profesor

Convenio: Llamaremos “Gestionar X” a todas las operaciones de adición, modificación ó eliminación del / de los elemento / s X.

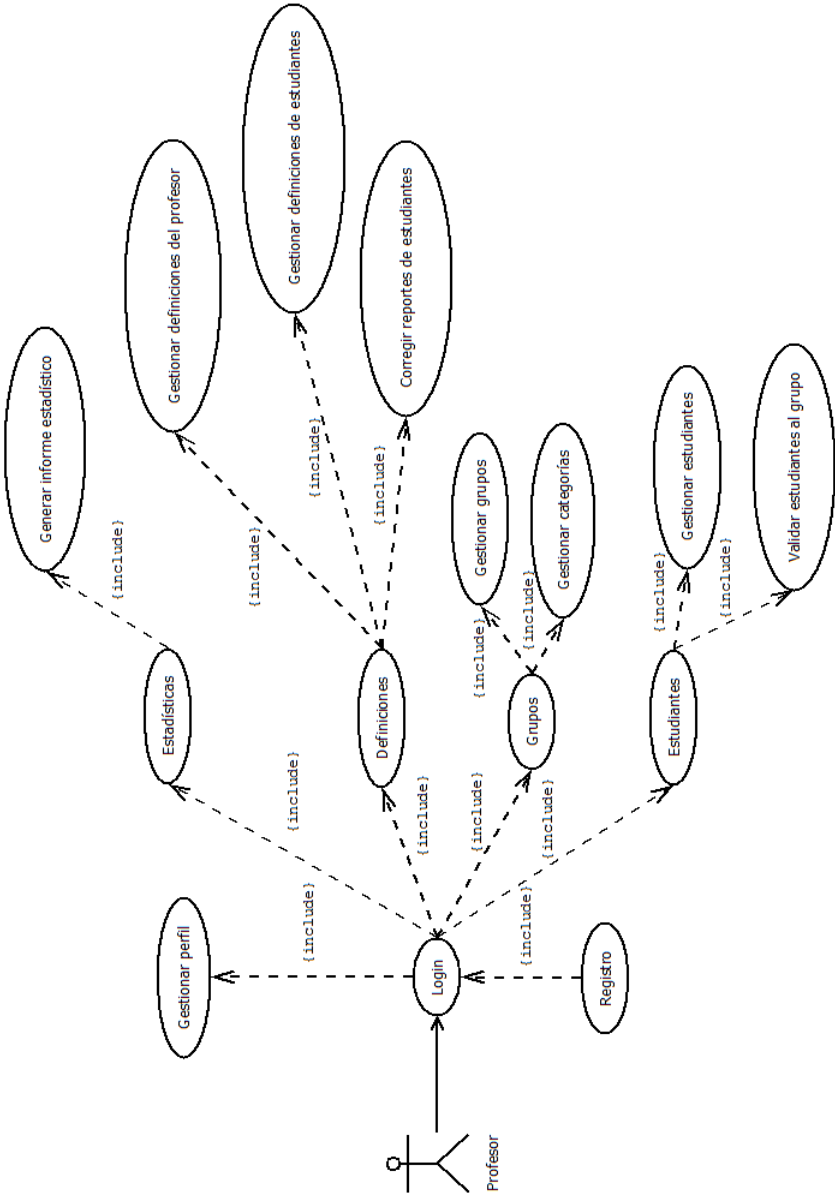


Figura 4.2: Análisis: Casos de uso del profesor

Descripción de los casos de uso

A continuación describimos los casos de uso del profesor en la web de nuestro sistema. Usaremos una plantilla y una descripción legible y entendible por un usuario no experto.

Caso de uso: Login**Caso de uso** Login

Descripción: Formulario de login de la web.

Actores: Usuario

Precondiciones: Usuario registrado y validado en el sistema.

Postcondiciones: Acceso a los menús de la web.

Escenario principal:

1. El sistema muestra el formulario de login.
2. El usuario introduce sus datos de acceso.
3. El sistema valida los datos y cambia a la vista de gestión.

Extensiones – Flujo alternativo

3a Los datos introducidos son incorrectos.

1. El sistema informa del error.
2. El usuario vuelve a introducir sus datos.

Caso de uso: Registro**Caso de uso** Registro

Descripción: Formulario de registro en el sistema.

Actores: Usuarios

Precondiciones: Usuario no existe en la base de datos.

Postcondiciones: Usuario registrado pendiente de validación por parte del administrador.

Escenario principal:

1. El sistema muestra el formulario de registro.
2. El usuario introduce los datos requeridos por el sistema.

3. El sistema registra en la base de datos al usuario y deja pendiente de validar por el administrador al usuario.

Caso de uso: Gestionar perfil

Caso de uso Gestionar perfil

Descripción: Formulario donde el docente puede modificar sus datos de acceso.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema.

Postcondiciones: Datos del usuario actualizados.

Escenario principal:

1. El sistema muestra un formulario con los datos actuales del profesor.
2. El usuario puede modificar cualquier dato.
3. El usuario hace click sobre el botón de “Guardar”.
4. El sistema valida y actualiza los nuevos datos.

Extensiones – Flujo alternativo

4a Algún dato no está en el formato correcto.

1. El sistema informa de dónde se ha producido el error de formato.
2. El usuario cambia los datos erróneos.
3. El usuario vuelve a hacer click sobre el botón “Guardar”.
4. El sistema valida los datos. {4. del flujo principal. }

Caso de uso: Estadísticas

Caso de uso Generar informe estadístico

Descripción: Formulario para la generación de un informe gráfico, en forma de tablas, o descarga de un CSV con los datos numéricos.

Actores: Usuario

Precondiciones: Usuario logueado y con un grupo activo.

Postcondiciones: Informe generado con los datos obtenidos del formulario previo.

Escenario principal:

1. El sistema muestra el selector de grupos.
2. El usuario selecciona el grupo del que generar la estadística.

3. El sistema muestra el formulario con un listado de usuarios del grupo previamente seleccionado.
4. El usuario introduce los datos para generar la página de resultados.
5. El usuario pulsa sobre el botón “Generar”.
6. El sistema genera o bien una gráfica, o una tabla, o prepara la descarga del CSV acorde a la elección del usuario.
7. El sistema muestra los resultados.

Caso de uso: Definiciones

Caso de uso Gestionar definiciones del profesor

Descripción: Conjunto de opciones para gestionar el contenido, por parte del profesor, del grupo.

Actores: Usuario

Precondiciones: Usuario logueado, con un grupo creado.

Postcondiciones: Adición, modificación o eliminación de contenido en el grupo.

Escenario principal:

1. El sistema muestra el submenú de gestión de definiciones.
2. El usuario selecciona la opción que desee realizar.
3. El sistema le muestra el selector de grupos.
4. El usuario selecciona el grupo sobre el que trabajar.
5. El sistema muestra el formulario correspondiente a la acción que desea realizar.

Caso de uso Gestionar definiciones de estudiantes

Descripción: Conjunto de opciones para gestionar el contenido, por parte de los estudiantes, del grupo.

Actores: Usuario

Precondiciones: Usuario logueado, grupo activo con usuarios jugando y habiendo añadido definiciones al sistema.

Postcondiciones: Modificación o eliminación del contenido aportado por los estudiantes al grupo.

Escenario principal:

1. El sistema muestra el submenú de gestión de definiciones, en este caso, de aportaciones al sistema por parte de los usuarios.
2. El usuario selecciona la opción que desee realizar.
3. El sistema le muestra el selector de grupos.
4. El usuario selecciona el grupo sobre el que desea realizar la operación.
5. El sistema le muestra el formulario correspondiente a la operación que desea realizar el usuario.
6. El usuario gestiona las aportaciones haciendo uso del formulario dado.

Caso de uso Corregir reportes de los estudiantes

Descripción: Submenú especial, con fines estadísticos, donde el profesor puede corregir los reportes de sus alumnos.

Actores: Usuario

Precondiciones: Usuario logueado, grupo creado con usuarios cliente y reportes realizados por éstos usuarios.

Postcondiciones: Reportes corregidos por el profesor, además de aportar información para un apartado de la generación de estadísticas.

Escenario principal:

1. El sistema muestra el selector de grupos.
2. El usuario selecciona el grupo a evaluar sus reportes.
3. El sistema muestra un listado con todos los reportes, su autor, la palabra que ha reportado, su artículo, definición y pista. Además de un apartado para corregir.
4. El usuario corrige las definiciones reportadas.
5. El usuario pulsa sobre el botón “Evaluar”.
6. El sistema registra la valoración del profesor en cada uno de los reportes corregidos.

Caso de uso: Grupos

Caso de uso Gestionar grupos

Descripción: Conjunto de opciones para gestionar los grupos, desde su creación, hasta su eliminación.

Actores: Usuario

Precondiciones: Usuario logueado.

Postcondiciones: Tener uno o varios grupos creados para su posterior uso en el juego.

Escenario principal:

1. El sistema muestra el submenú de gestión de los grupos.
2. El usuario selecciona la operación deseada.
3. El sistema muestra el formulario correspondiente a la acción que ha seleccionado el usuario.
4. El usuario hace uso del formulario para realizar la operación.
5. El usuario pulsa sobre el botón “Enviar” o “Guardar”.

Caso de uso Gestionar categorías

Descripción: Conjunto de operaciones para la gestión de las categorías de palabras de cada grupo que pertenezca al profesor.

Actores: Usuario

Precondiciones: Usuario logueado con al menos un grupo creado.

Postcondiciones: Modificación de las categorías del grupo seleccionado.

Escenario principal:

1. El sistema el submenú de gestión de categorías de los grupos.
2. El usuario selecciona la operación deseada.
3. El sistema muestra el selector de grupos.
4. El usuario selecciona el grupo sobre el que realizar la operación.
5. El sistema muestra el formulario correspondiente a la operación escogida por el usuario.
6. El usuario hace uso del formulario para gestionar las categorías del grupo seleccionado.

Caso de uso: Estudiantes

Caso de uso Gestionar estudiantes

Descripción: Conjunto de operaciones para la gestión de los estudiantes de un grupo seleccionado en el sistema.

Actores: Usuario

Precondiciones: Usuario logueado, con al menos un grupo creado y usuarios cliente en ese grupo.

Postcondiciones: Gestión de los datos de los usuarios cliente de un grupo seleccionado.

Escenario principal:

1. El sistema muestra un submenú con las operaciones posibles a realizar.
2. El usuario selecciona una de esas operaciones.
3. El sistema muestra el selector de grupos.
4. El usuario selecciona el grupo sobre el que realizar la operación.
5. El sistema muestra el formulario correspondiente a la operación seleccionada.
6. El usuario hace uso del formulario para gestionar la información de sus estudiantes.

Caso de uso Validar estudiantes

Descripción: Formulario donde el usuario da acceso a los clientes interesados en jugar en uno de sus grupos.

Actores: Usuario

Precondiciones: Usuario logueado, con al menos un grupo creado y solicitudes de acceso al grupo pendientes.

Postcondiciones: Usuarios cliente validados en el grupo y con acceso al juego.

Escenario principal:

1. El sistema muestra el selector de grupos.
2. El usuario selecciona el grupo sobre el que realizar la operación de validar.
3. El sistema muestra un listado con todos los usuarios cliente esperando a recibir validación del usuario.
4. El usuario selecciona todos aquellos estudiantes a los que dar acceso.
5. El usuario pulsa sobre el botón “Validar”.

4.3.3. Casos de uso del administrador

Convenio: Llamaremos “Gestionar X” a todas las operaciones de adición, modificación ó eliminación del / de los elemento / s X.

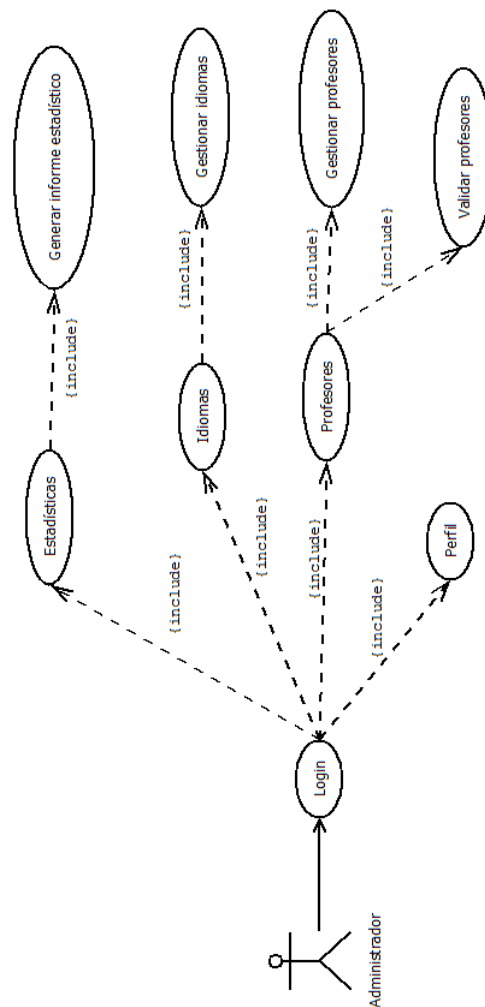


Figura 4.3: Análisis: Casos de uso del administrador

Descripción de los casos de uso

A continuación describimos los casos de uso del usuario del cliente de nuestro sistema. Usaremos una plantilla y una descripción legible y entendible por un usuario no experto.

Caso de uso: Login

Caso de uso Login

Descripción: Formulario de login de la web.

Actores: Usuario

Precondiciones: Usuario registrado y validado en el sistema.

Postcondiciones: Acceso a los menús de la web.

Escenario principal:

1. El sistema muestra el formulario de login.
2. El usuario introduce sus datos de acceso.
3. El sistema valida los datos y cambia a la vista de gestión.

Extensiones – Flujo alternativo

- 3a** Los datos introducidos son incorrectos.
1. El sistema informa del error.
 2. El usuario vuelve a introducir sus datos.

Caso de uso: Perfil

Caso de uso Perfil

Descripción: Formulario donde el administrador puede modificar sus datos de acceso.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema.

Postcondiciones: Datos del usuario actualizados.

Escenario principal:

1. El sistema muestra un formulario con los datos actuales del profesor.
2. El usuario puede modificar cualquier dato.
3. El usuario hace click sobre el botón de “Guardar”.
4. El sistema valida y actualiza los nuevos datos.

Extensiones – Flujo alternativo

- 4a** Algún dato no está en el formato correcto.
1. El sistema informa de dónde se ha producido el error de formato.
 2. El usuario cambia los datos erróneos.
 3. El usuario vuelve a hacer click sobre el botón “Guardar”.
 4. El sistema valida los datos. {4. del flujo principal. }

Caso de uso: Estadísticas

Caso de uso Generar informe estadístico

Descripción: Formulario para la generación de un informe gráfico, en forma de tablas, o descarga de un CSV con los datos numéricos.

Actores: Usuario

Precondiciones: Usuario logueado como administrador

Postcondiciones: Informe generado con los datos obtenidos del formulario previo.

Escenario principal:

1. El sistema muestra el selector de grupos. Ésta vez muestra todos los grupos del sistema.
2. El usuario selecciona el grupo del que generar la estadística.
3. El sistema muestra el formulario con un listado de usuarios del grupo previamente seleccionado.
4. El usuario introduce los datos para generar la página de resultados.
5. El usuario pulsa sobre el botón “Generar”.
6. El sistema genera o bien una gráfica, o una tabla, o prepara la descarga del CSV acorde a la elección del usuario.
7. El sistema muestra los resultados desglosados en los resultados de los algoritmos para el Test A-B del sistema.

Caso de uso: Idiomas

Caso de uso Gestionar idiomas

Descripción: Conjunto de operaciones donde el administrador gestiona los idiomas jugables del sistema.

Actores: Usuario

Precondiciones: Usuario logueado en el sistema.

Postcondiciones: Datos de los idiomas disponibles en el sistema actualizados.

Escenario principal:

1. El sistema muestra el submenú de gestión de los idiomas del sistema.
2. El usuario selecciona la operación deseada.
3. El sistema muestra el formulario asociado a la operación seleccionada.
4. El usuario hace uso del formulario para realizar la operación deseada.
5. El usuario pulsa sobre el botón “Guardar” o “Enviar”.

Caso de uso: Profesores**Caso de uso** Gestionar profesores

Descripción: Conjunto de operaciones para la gestión de los profesores en el sistema.

Actores: Usuario

Precondiciones: Usuario logueado.

Postcondiciones: Gestión de los datos de los profesores.

Escenario principal:

1. El sistema muestra un submenú con las operaciones posibles a realizar.
2. El usuario selecciona una de esas operaciones.
3. El sistema muestra el formulario correspondiente a la operación seleccionada.
4. El usuario hace uso del formulario para gestionar la información de los profesores.

Caso de uso Validar profesores

Descripción: Formulario donde el administrador da acceso a los profesores para poder utilizar la web.

Actores: Usuario

Precondiciones: Usuario logueado y solicitudes de acceso a la plataforma pendientes.

Postcondiciones: Profesores validados en el sistema y con acceso a la web.

Escenario principal:

1. El sistema muestra un listado con todos los profesores esperando a recibir validación del usuario.
2. El usuario selecciona todos aquellos profesores a los que dar acceso.
3. El usuario pulsa sobre el botón “Validar”.

4.4. Modelo conceptual de datos

Este apartado del análisis sirve para especificar los requisitos del sistema y las relaciones estáticas que existen entre ellos.

Para este fin se utiliza como herramienta los diagramas de clase. En estos diagramas se representan las clases de objetos, las asociaciones entre dichas clases, los atributos que componen las clases y las relaciones de integridad.

4.4.1. Diagrama Entidad-Relación

En las siguientes imágenes podemos ver los diagramas ER que representan las entidades relevantes del sistema de información implementado, así como sus interrelaciones y propiedades:

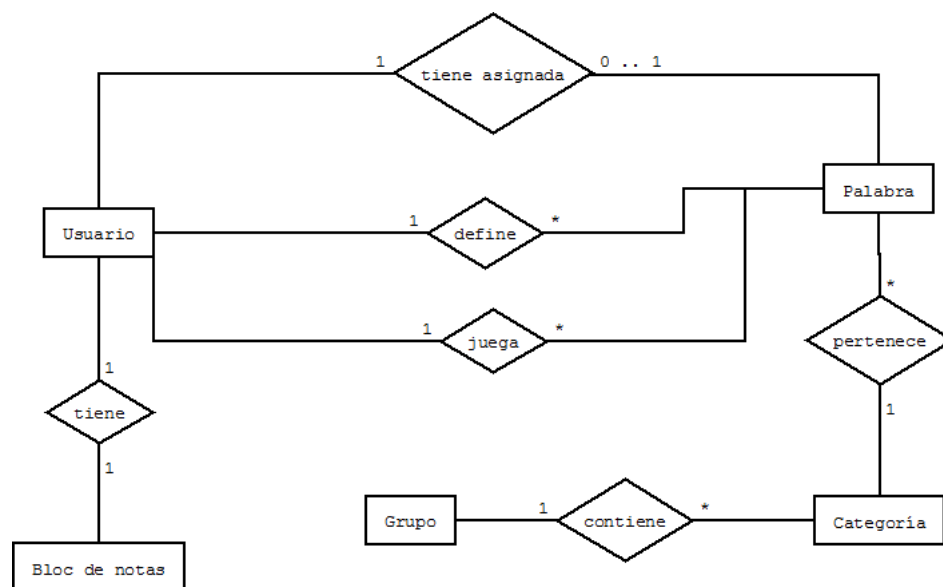


Figura 4.4: Análisis: Diagrama ER del cliente.

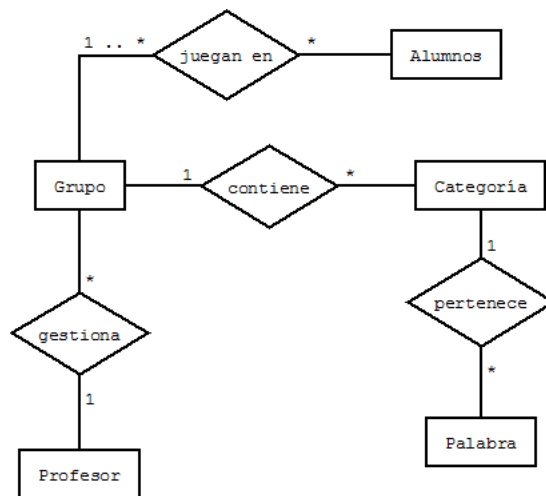


Figura 4.5: Análisis: Diagrama ER de la web del profesor.

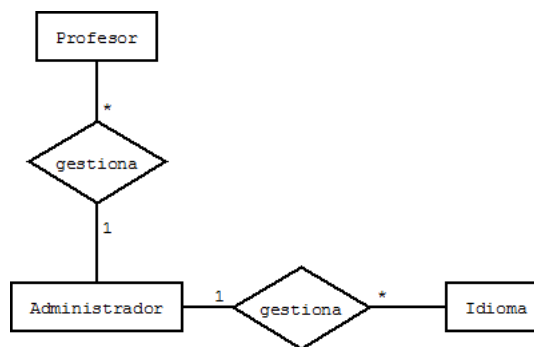


Figura 4.6: Análisis: Diagrama ER del administrador.

4.5. Modelo de comportamiento

El modelo de comportamiento especifica como debe actuar el sistema. El sistema es el que engloba todos los objetos, y el modelo consta de dos partes:

- Diagramas de secuencias del sistema: muestran la secuencia de eventos entre el usuario y el sistema.
- Contrato de las operaciones del sistema: describen el efecto que producen las operaciones en el sistema.

4.5.1. Diagramas de secuencia y contrato de las operaciones

No todos los posibles diagramas de secuencia aparecerán, nos centraremos en los más importantes, los que implican algún tipo de cambio en el sistema. Para evitar con-

tenidos duplicados, se omitirán las operaciones que hayan sido explicadas con anterioridad o que sean muy similares a las anteriores.

Caso de uso: Menú principal

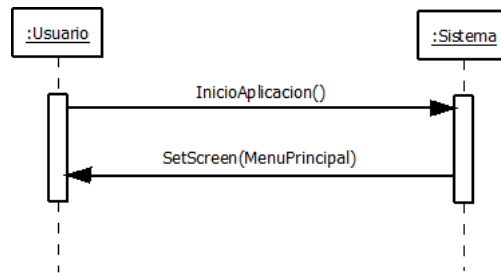


Figura 4.7: Análisis: Diagrama de flujo del menú principal.

Operación InicioAplicacion()

Actores Usuario, Sistema

Responsabilidades Inicia la aplicación.

Precondiciones Ninguna.

Postcondiciones Aplicación iniciada.

Operación SetScreen(MenuPrincipal)

Actores Sistema

Responsabilidades Cambiar la screen a mostrar a la clase screen del argumento.

Precondiciones Ninguna.

Postcondiciones Muestra la screen del argumento.

Caso de uso: Login

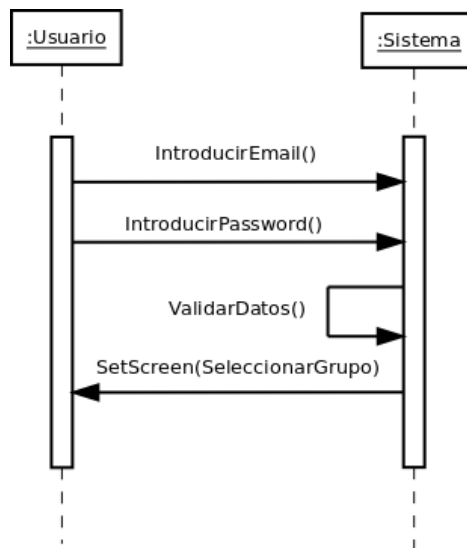


Figura 4.8: Análisis: Diagrama de flujo del escenario principal.

Operación IntroducirEmail()

Actores Usuario

Responsabilidades El usuario introduce su email para validar.

Precondiciones Usuario registrado en el sistema.

Postcondiciones Email del usuario guardado de forma temporal para su validación.

Operación IntroducirPassword()

Actores Usuario

Responsabilidades El usuario introduce su clave única en el sistema.

Precondiciones Usuario registrado.

Postcondiciones Password del usuario guardado de forma temporal para su validación.

Operación ValidarDatos()

Actores Sistema

Responsabilidades Compara los datos introducidos en los campos del formulario con los almacenados en la base de datos y da una respuesta.

Precondiciones Datos introducidos en los formularios pertinentes.

Postcondiciones Respuesta del sistema al usuario por los datos introducidos en el formulario.

Flujo alternativo: Login

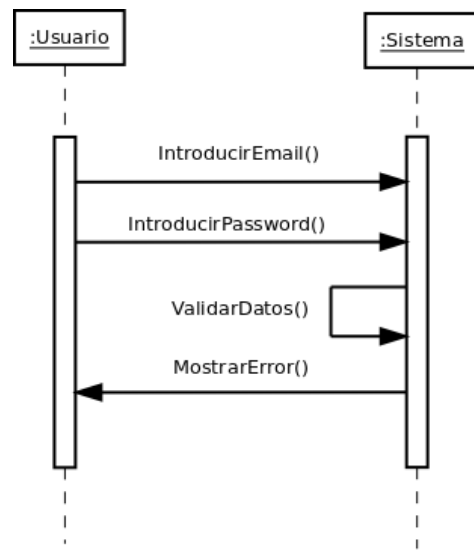


Figura 4.9: Análisis: Escenario alternativo al principal del caso Login.

Operación MostrarError()

Actores Sistema

Responsabilidades Muestra una ventana con la información del error producido.

Precondiciones Sistema valida la información introducida.

Postcondiciones Ventana emerge con el mensaje de error.

Caso de uso: Registro

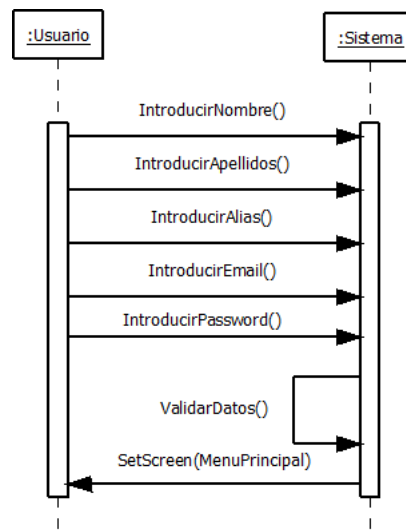


Figura 4.10: Análisis: Diagrama de flujo de registro en el sistema.

Operación Validar datos()**Actores** Sistema**Responsabilidades** El sistema comprueba, a través de expresiones regulares, que los datos introducidos no presentan errores para su posterior almacenamiento en la base de datos.**Precondiciones** Campos rellenos con los datos a validar.**Postcondiciones** Datos verificados y enviados al servidor para almacenarlos en la base de datos.

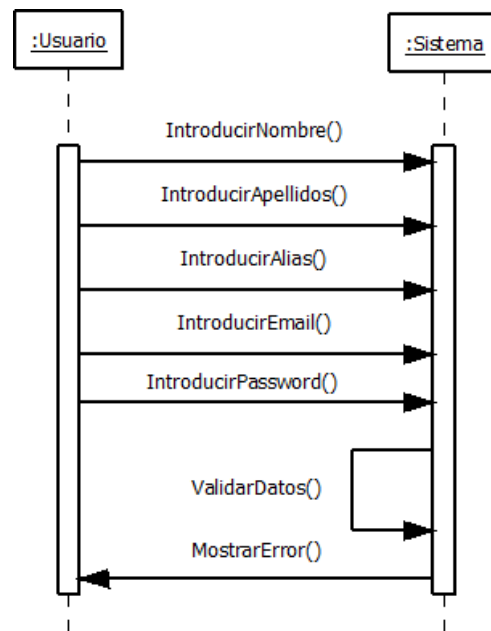


Figura 4.11: Análisis: Escenario alternativo al caso de uso principal de registro.

Operación Validar datos()

Actores Sistema

Responsabilidades El sistema comprueba, a través de expresiones regulares, que los datos introducidos no presentan errores para su posterior almacenamiento en la base de datos.

Precondiciones Campos rellenos con los datos a validar.

Postcondiciones Datos erróneos en los campos indicados en el error.

Caso de uso: Créditos

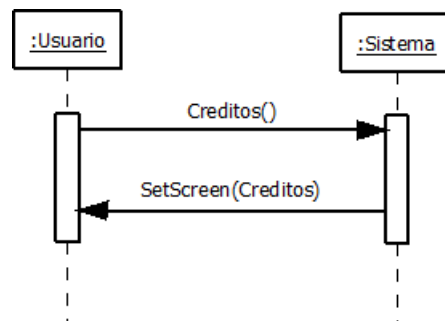


Figura 4.12: Análisis: Diagrama de flujo de los créditos del sistema.

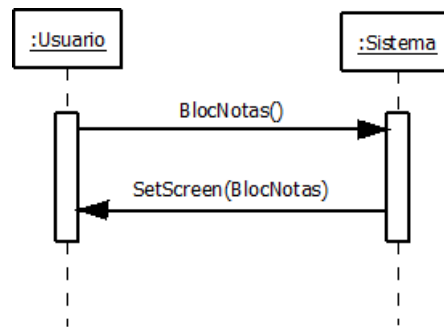
Operación Creditos()**Actores** Usuario**Responsabilidades** Se encarga de mostrar la ventana con los créditos del equipo de desarrollo de la aplicación y de todos los implicados en el proyecto.**Precondiciones** Ninguna.**Postcondiciones** Se muestra la pantalla de créditos.**Caso de uso: Bloc de notas**

Figura 4.13: Análisis: Diagrama de flujo del bloc de notas

Operación BlocNotas()**Actores** Usuario**Responsabilidades** Indica al sistema que deseamos ver el bloc de notas.**Precondiciones** Ninguna**Postcondiciones** Se muestra el bloc de notas con sus componentes.**Caso de uso: Seleccionar grupo**

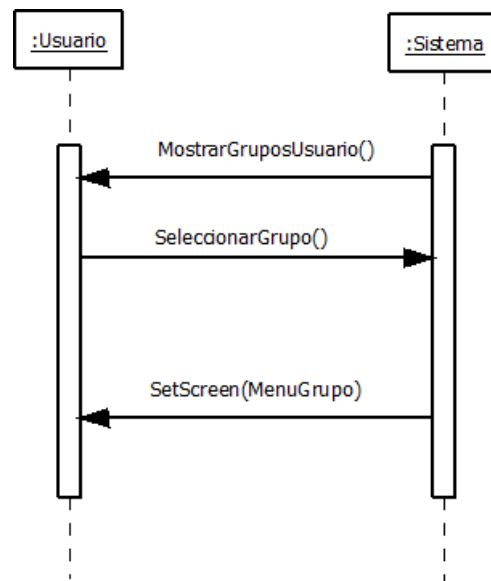


Figura 4.14: Análisis: Diagrama de flujo de seleccionar grupo

Operación SeleccionarGrupo()**Actores** Usuario**Responsabilidades** Operación que guarda las credenciales del grupo necesarias para el resto de operaciones y cambia la pantalla al menú del grupo.**Precondiciones** Usuario logueado y con al menos un grupo válido.**Postcondiciones** Cambio a la ventana de menú del grupo.**Caso de uso: Unirse a grupo**

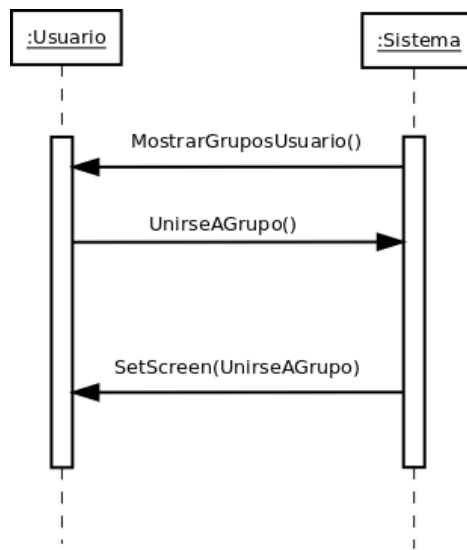


Figura 4.15: Análisis: Diagrama de flujo de unirse a grupo

Operación UnirseAGrupo()**Actores** Usuario**Responsabilidades** Seleccionar la pantalla de unirse a un grupo.**Precondiciones** Usuario logueado.**Postcondiciones** Se muestra la pantalla de enviar solicitudes a grupos.**Caso de uso: Enviar solicitudes**

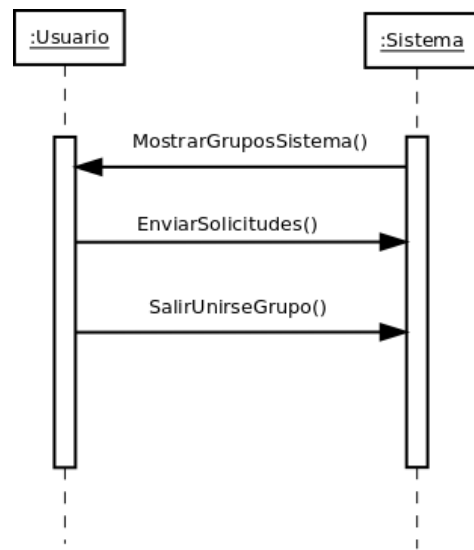


Figura 4.16: Análisis: Diagrama de flujo de enviar solicitudes

Operación EnviarSolicitudes()**Actores** Usuario**Responsabilidades** Enviar las solicitudes por HTTP.**Precondiciones** Usuario logueado, grupo seleccionado para enviar solicitud.**Postcondiciones** Solicitudes aparecen pendientes de validar en la base de datos.**Operación** SalirUnirseGrupo()**Actores** Usuario**Responsabilidades** Salir al menú de selección de grupos.**Precondiciones** Usuario logueado.**Postcondiciones** Salir a la pantalla anterior.**Caso de uso: Menú de grupo**

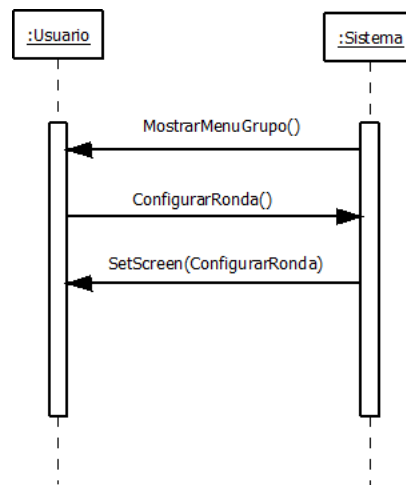


Figura 4.17: Análisis: Casos de uso del profesor

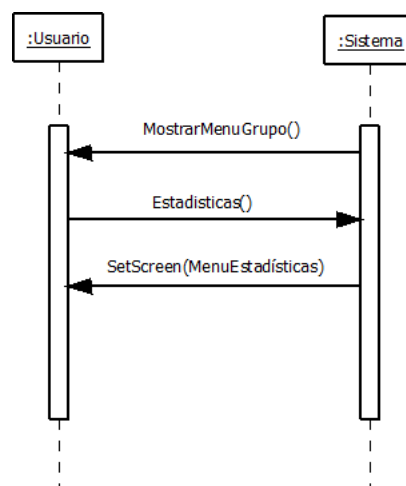


Figura 4.18: Análisis: Casos de uso del profesor

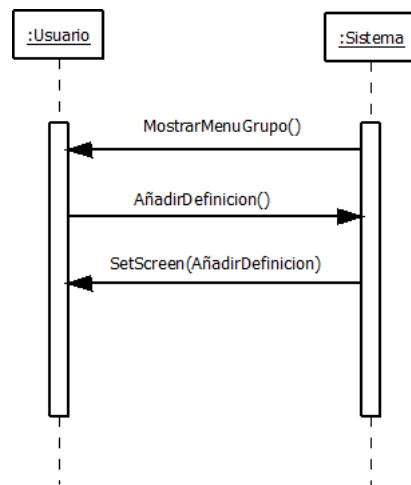


Figura 4.19: Análisis: Casos de uso del profesor

Operación ConfigurarRonda()**Actores** Usuario**Responsabilidades** Operación que cambia de ventana a la de configurar una ronda de juego.**Precondiciones** Usuario logueado. Grupo seleccionado.**Postcondiciones** Cambia la pantalla a la configuración de la ronda.**Operación** Estadísticas()**Actores** Usuario**Responsabilidades** Cambia la pantalla al menú de estadísticas.**Precondiciones** Usuario logueado. Grupo seleccionado.**Postcondiciones** Cambia la pantalla al menú de estadísticas.**Operación** AñadirDefinicion()**Actores** Usuario**Responsabilidades** Cambia la pantalla a la de añadir una nueva definición.**Precondiciones** Usuario logueado, grupo seleccionado y habiendo cumplido la condición para desbloquear ésta operación.**Postcondiciones** Da acceso a la pantalla para añadir una nueva definición al sistema.**Caso de uso: Añadir definición**

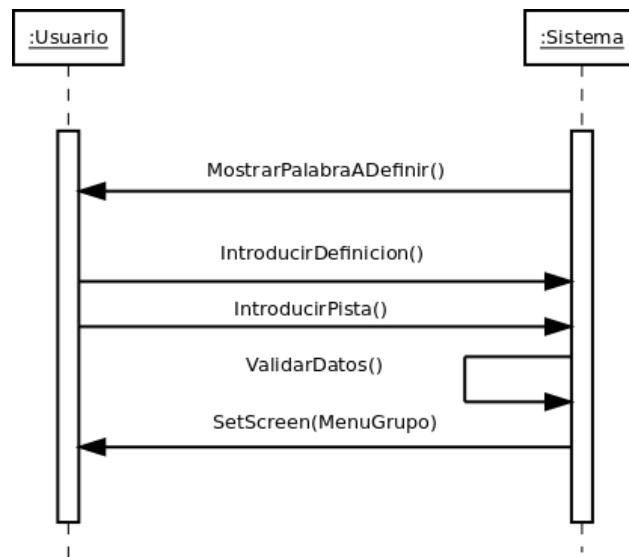


Figura 4.20: Análisis: Diagrama de flujo de añadir definición.

Operación MostrarPalabraADefinir()**Actores** Sistema**Responsabilidades** Muestra al usuario la palabra, con su artículo si procede, la categoría y el nivel del que tiene que proporcionar una definición.**Precondiciones** Conexión a base de datos exitosa y obtención de una palabra al azar de entre la lista de las menos definidas en el grupo.**Postcondiciones** Muestra los datos en pantalla.**Caso de uso:** Logros

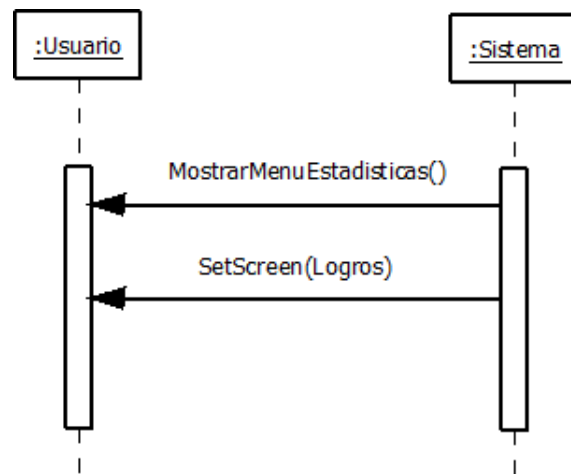


Figura 4.21: Análisis: Diagrama de flujo de la pantalla de logros.

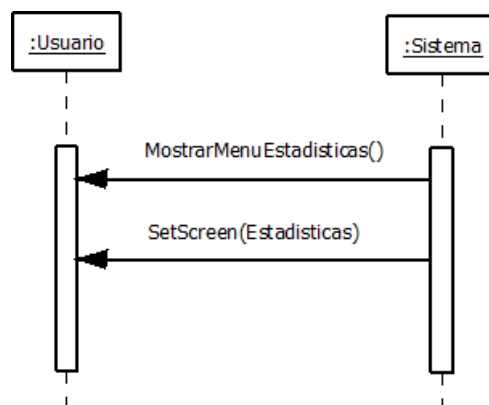
Operación Logros()**Actores** Usuario**Responsabilidades** Operación que cambia la pantalla a la de logros del usuario.**Precondiciones** Usuario logueado, grupo seleccionado.**Postcondiciones** Cambia la pantalla a la de logros.**Caso de uso: Estadísticas de usuario**

Figura 4.22: Análisis: Diagrama de flujo de las estadísticas de usuario.

Operación Estadísticas()**Actores** Usuario**Responsabilidades** Operación que cambia la pantalla a la de estadísticas del usuario.**Precondiciones** Usuario logueado, grupo seleccionado.

Postcondiciones Cambia la pantalla a la de estadísticas.

Caso de uso: Configurar partida

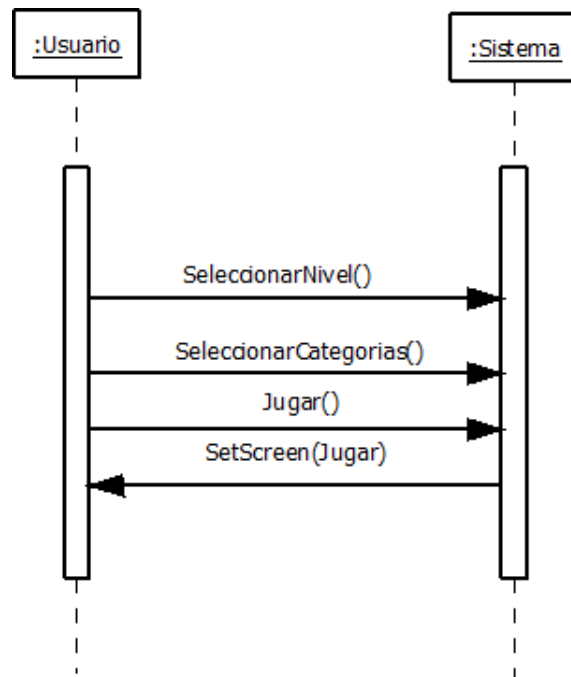


Figura 4.23: Análisis: Diagrama de flujo de configurar una ronda.

Operación SeleccionarNivel()

Actores Usuario

Responsabilidades Seleccionar el nivel de la ronda.

Precondiciones Usuario logueado, grupo seleccionado.

Postcondiciones Nivel configurado para la ronda.

Operación SeleccionarCategorias()

Actores Usuario

Responsabilidades Muestra un menú donde podemos escoger cualquier conjunto de categorías para la ronda.

Precondiciones Usuario logueado, grupo seleccionado, grupo con categorías y definiciones.

Postcondiciones Categorías configuradas para la ronda.

Operación Jugar()

Actores Usuario

Responsabilidades Comienza el juego.

Precondiciones Usuario logueado, grupo seleccionado, ronda configurada.

Postcondiciones Cambia a la pantalla de juego con la configuración previamente seleccionada.

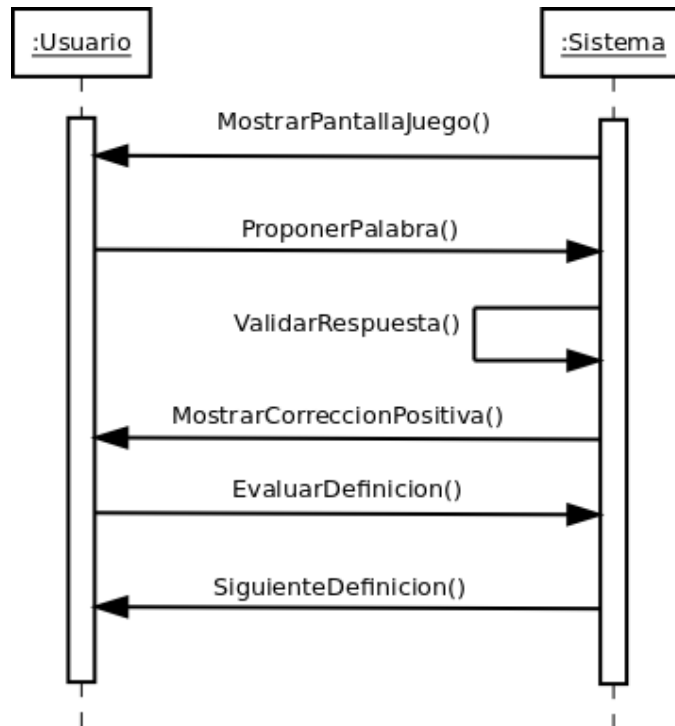
Caso de uso: Jugar

Figura 4.24: Análisis: Diagrama de flujo de caso principal de juego

Operación ProponerPalabra()**Actores** Usuario**Responsabilidades** El usuario introduce una palabra para su posterior validación como respuesta.**Precondiciones** Partida generada, datos cargados correctamente.**Postcondiciones** Enviar la palabra al engine para validar.**Operación EvaluarDefinicion()****Actores** Usuario**Responsabilidades** Evaluar la definición para otorgarle una puntuación.**Precondiciones** Mostrada la ventana de resultado.**Postcondiciones** Puntuación enviada al servidor.

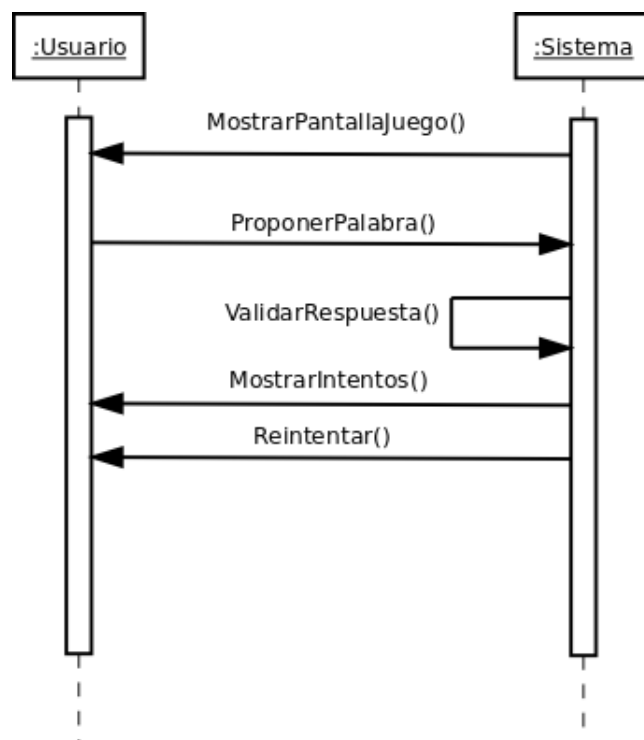


Figura 4.25: Análisis: Diagrama de flujo de caso alternativo a jugar.

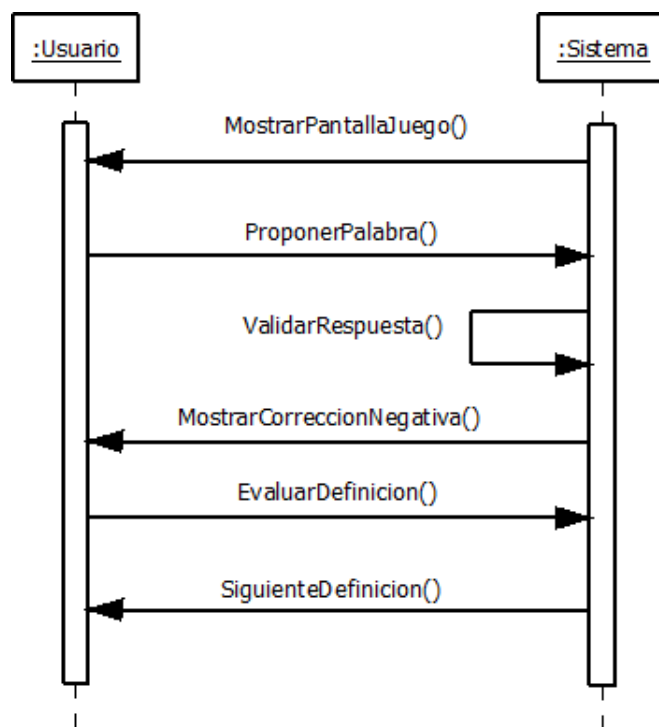


Figura 4.26: Análisis: Diagrama de flujo de caso alternativo a jugar en caso de error.

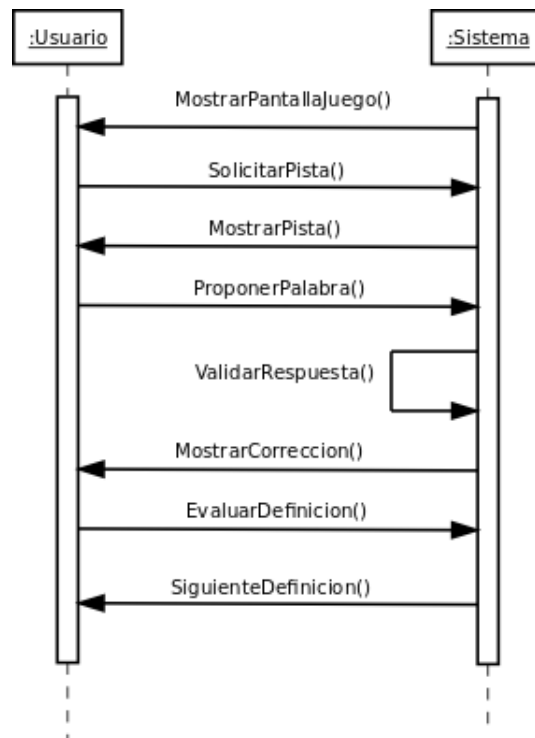


Figura 4.27: Análisis: Diagrama de flujo de juego cuando se usa pista.

Operación SolicitarPista()**Actores** Usuario**Responsabilidades** Muestra una ventana emergente con la pista correspondiente a la definición jugándose.**Precondiciones** Haber fallado la palabra a adivinar al menos una vez.**Postcondiciones** Muestra la ventana emergente con la pista.

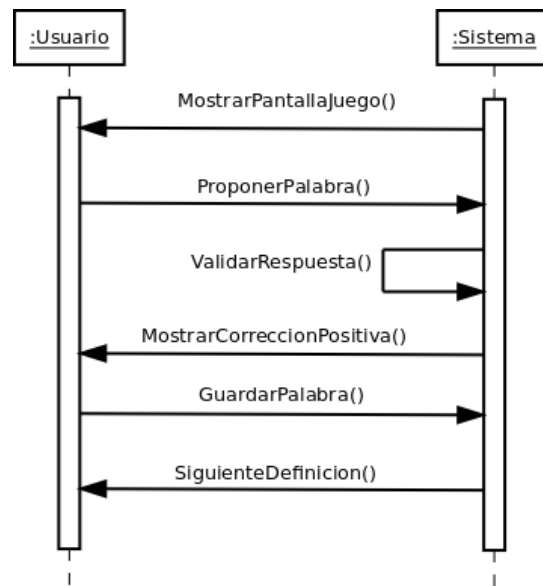


Figura 4.28: Análisis: Diagrama de flujo juego al guardar palabra en el bloc

Operación GuardarPalabra()

Actores Usuario

Responsabilidades Guarda la palabra en el bloc de notas.

Precondiciones Palabra jugada, acertada o no, ventana de evaluación mostrada.

Postcondiciones La palabra queda almacenada en un archivo local del bloc de notas.

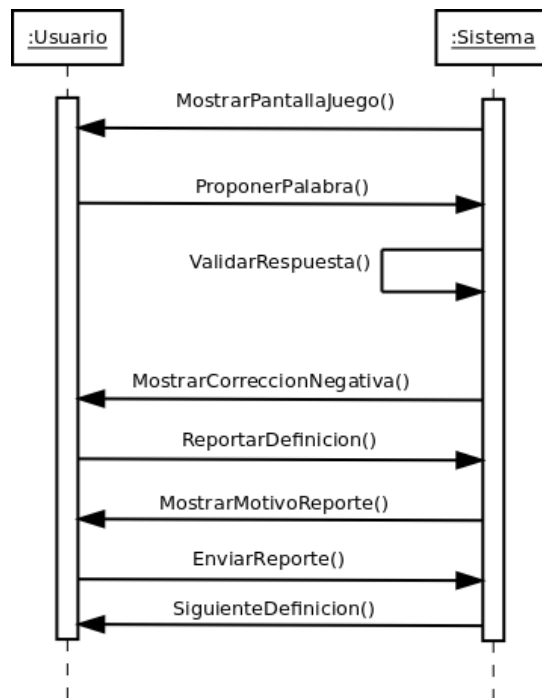


Figura 4.29: Análisis: Diagrama de flujo al reportar una definición

Operación ReportarDefinicion()**Actores** Usuario**Responsabilidades** Muestra la ventana con el motivo del reporte.**Precondiciones** Definición jugada, acertada o no. Pantalla de evaluación mostrada.**Postcondiciones** Motivo del reporte seleccionado esperando a ser enviado al servidor.**Operación EnviarReporte()****Actores** Usuario**Responsabilidades** Envía el motivo del reporte al servidor.**Precondiciones** Definición jugada, acertada o no. Pantalla de evaluación mostrada.
Motivo del reporte seleccionado.**Postcondiciones** Almacenado en la base de datos el motivo del reporte.**Caso de uso: Gestión de los alumnos**

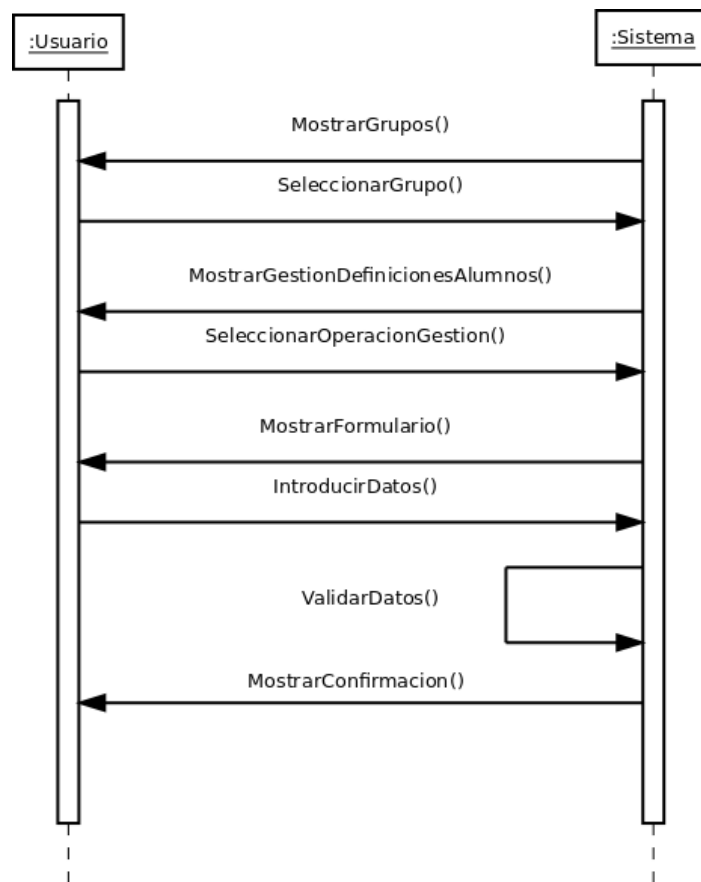


Figura 4.30: Análisis: Diagrama de flujo de operaciones de gestión de alumnos

Operación SeleccionarGrupo()

Actores Usuario

Responsabilidades Selecciona el grupo sobre el que realizar cualquier operación.

Precondiciones Usuario logueado, algún grupo creado.

Postcondiciones El grupo queda seleccionado.

Operación SeleccionarOperacionGestion()

Actores Usuario

Responsabilidades Submenú donde se selecciona la operación de gestión a llevar a cabo.

Precondiciones Grupo seleccionado.

Postcondiciones Operación de gestión seleccionada.

Operación IntroducirDatos()

Actores Usuario

Responsabilidades Dependiendo de la operación, saldrá un formulario diferente que hay que rellenar con datos o seleccionarlos.

Precondiciones Grupo seleccionado y operación seleccionada. Además de formulario cargado.

Postcondiciones Datos introducidos para llevar a cabo la gestión.

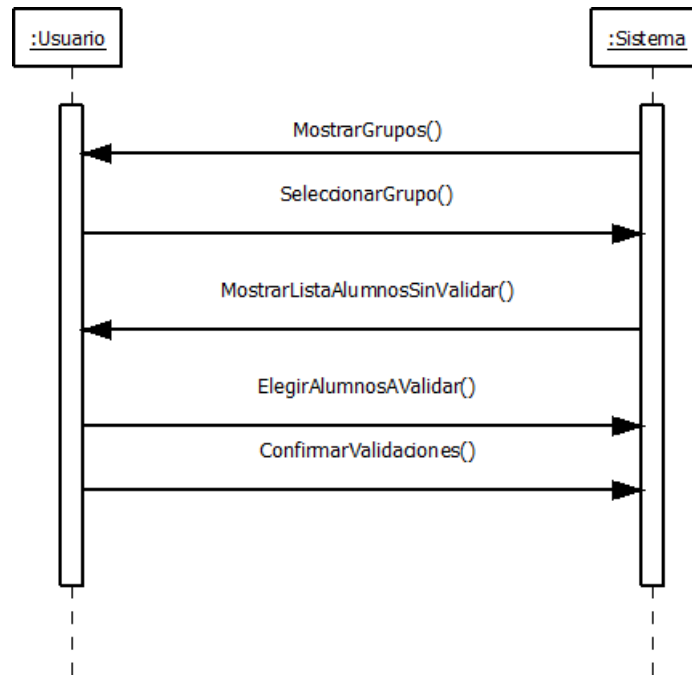
Caso de uso: Validar alumnos

Figura 4.31: Análisis: Diagrama de flujo de validación de alumnos

Operación ElegirAlumnosAValidar()**Actores** Usuario**Responsabilidades** Operación que guarda las identificaciones de los alumnos para su posterior procesamiento.**Precondiciones** Grupo creado con solicitudes pendientes.**Postcondiciones** Identificaciones almacenadas para su procesamiento posterior.**Operación ConfirmarValidaciones()****Actores** Usuario**Responsabilidades** Mostrear un mensaje de confirmación antes de proceder con la validación de los usuarios que soliciten acceso al grupo.**Precondiciones** Usuarios seleccionados pendientes de validación.**Postcondiciones** Los usuarios seleccionados tendrán acceso al grupo de forma inmediata.**Caso de uso: Corregir definiciones**

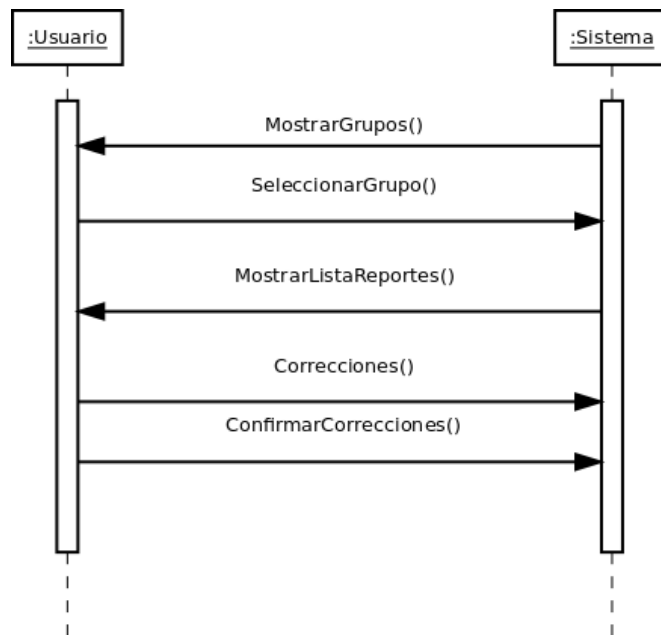


Figura 4.32: Análisis: Diagrama de flujo de corrección de definiciones

Operación Correcciones()**Actores** Usuario**Responsabilidades** Método que almacena las correcciones de los reportes realizados a través del cliente.**Precondiciones** Existen reportes generados desde el cliente.**Postcondiciones** Almacenadas las correcciones para su posterior actualización en la base de datos.**Operación ConfirmarCorrecciones()****Actores** Usuario**Responsabilidades** Muestra una ventana de confirmación antes de realizar la operación de actualización en la base de datos.**Precondiciones** Correcciones preparadas para su envío a la base de datos.**Postcondiciones** Correcciones actualizadas en el sistema.**Caso de uso: Generar estadísticas**

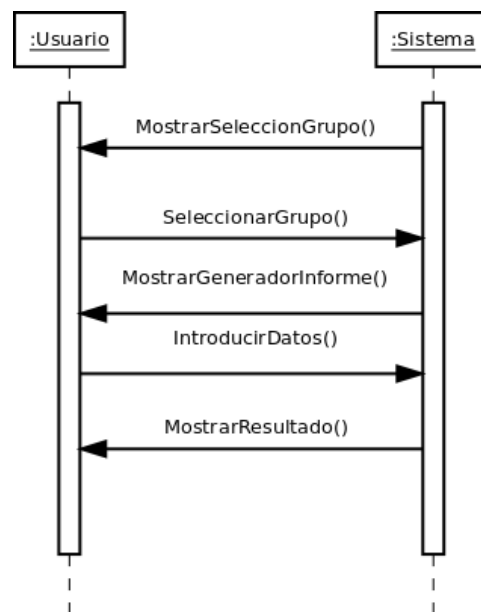


Figura 4.33: Análisis: Diagrama de flujo del generador de informes.

Caso de uso: Gestionar profesores

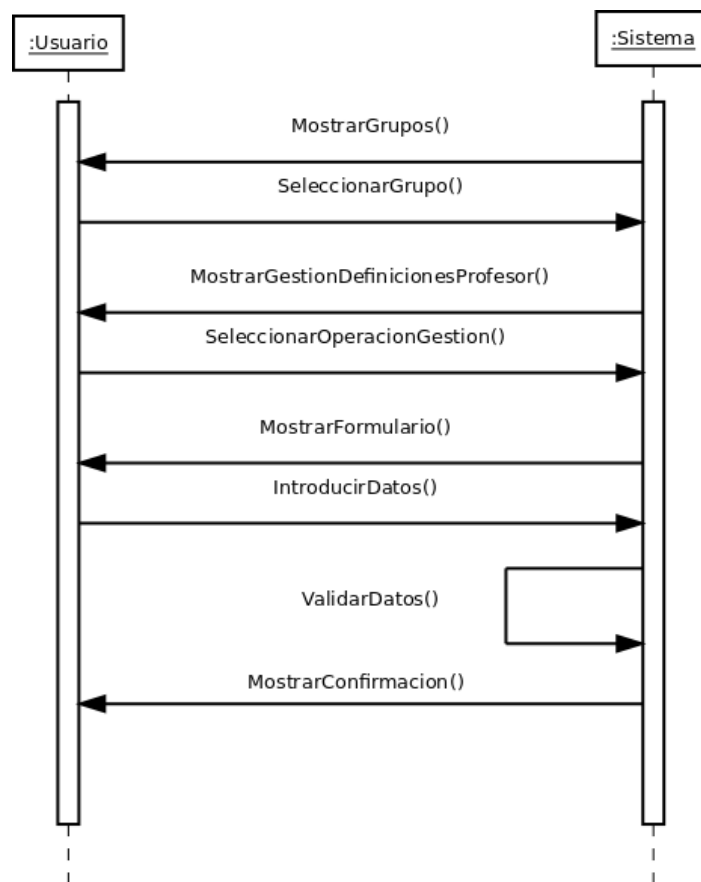


Figura 4.34: Análisis: Diagrama de flujo de la gestión de los profesores.

El proceso de descripción del diseño es mucho más sencillo una vez que ya se ha especificado que hace el sistema en el capítulo anterior. También decir, que al igual que el análisis del sistema, el diseño no contempla muchos detalles del sistema final, sólo una idea orientativa de como se implementará el sistema.

5.1. Arquitectura

En esta sección se define la arquitectura general del sistema de información, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

5.1.1. Física

Los elementos hardware necesarios para la ejecución de nuestro sistema se dividen en:

- Dispositivo móvil
- PC

La app se puede ejecutar desde ambos elementos, cumpliendo un mínimo de requisitos para su correcta ejecución:

- Procesador: Mononúcleo con 1GHz de velocidad
- Memoria RAM: 1GB
- Memoria secundaria: Mínimo 10 MB de espacio

La web necesita de un PC capaz de ejecutar un navegador web actualizado, es recomendable el uso de Google Chrome para una correcta visualización de las gráficas.

5.1.2. Lógica

La arquitectura lógica comprende todo aquel software necesario para la elaboración y ejecución del sistema. A continuación se explican los distintos elementos de los que se compone el sistema:

Git

Se ha escogido a Git como sistema de control de versiones, muy usado en grandes proyectos, como, por ejemplo, el núcleo de Linux. Se trata de un software libre bajo licencia GPL.

Haciendo uso de Git, subiremos la implementación de nuestro proyecto a un repositorio al que podremos acceder a él de forma remota a través del PC, además de poder restaurar a versiones previas del proyecto, en caso de producirse errores.

El repositorio escogido es GitHub, ya que, además de ser un repositorio libre, es el que mejores prestaciones ofrece, algunas de sus características son:

- Una wiki, donde recogemos la información de nuestro proyecto y que siempre tendremos actualizado.
- Sistema de seguimiento de problemas, donde cualquier participante del proyecto puede recalcar errores o problemas para que el resto de sus compañeros lo tenga presente.
- Revisión de código, donde se pueden añadir anotaciones en cualquier parte del código.
- Ramas donde comparar el progreso de las diferentes partes del proyecto.

Gradle

Se trata de una herramienta para automatizar la construcción de proyectos, de uso obligatorio con el framework del proyecto. Disponemos también de herramientas para compilar, testear y desplegar nuestro proyecto.

LibGDX

Se ha usado LibGDX como framework para el desarrollo del cliente móvil. Se trata de un framework multiplataforma, que soporta actualmente el desarrollo para Windows, Linux, Mac OSX, Android, iOS, HTML5 y Blackberry.

El framework permite, de una forma sencilla, realizar el código sólo una vez y luego exportarlo a las plataformas que soporta, haciendo uso de la herramienta antes mencionada, Gradle, para desplegar el proyecto.

El lenguaje con el que trabaja el framework es Java.

Android SDK

Aunque no de forma directa, necesitamos el SDK de Android para la exportación del proyecto al instalable para dicha plataforma.

No es necesario descargar el ADT de Android para la compilación del proyecto. La actualización del SDK se ha llevado a cabo desde el Android SDK Manager.

MySQL

Se trata del sistema de gestión de bases de datos más usado y, por lo tanto, más popular, entre sus características destacamos:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

En el proyecto, se ha usado para almacenar la información de usuarios, contenido y gestiones de la plataforma.

5.2. Diseño físico

Tras el primer planteamiento sobre el contenido que almacenará la aplicación, se llega al siguiente diagrama lógico para la base de datos:

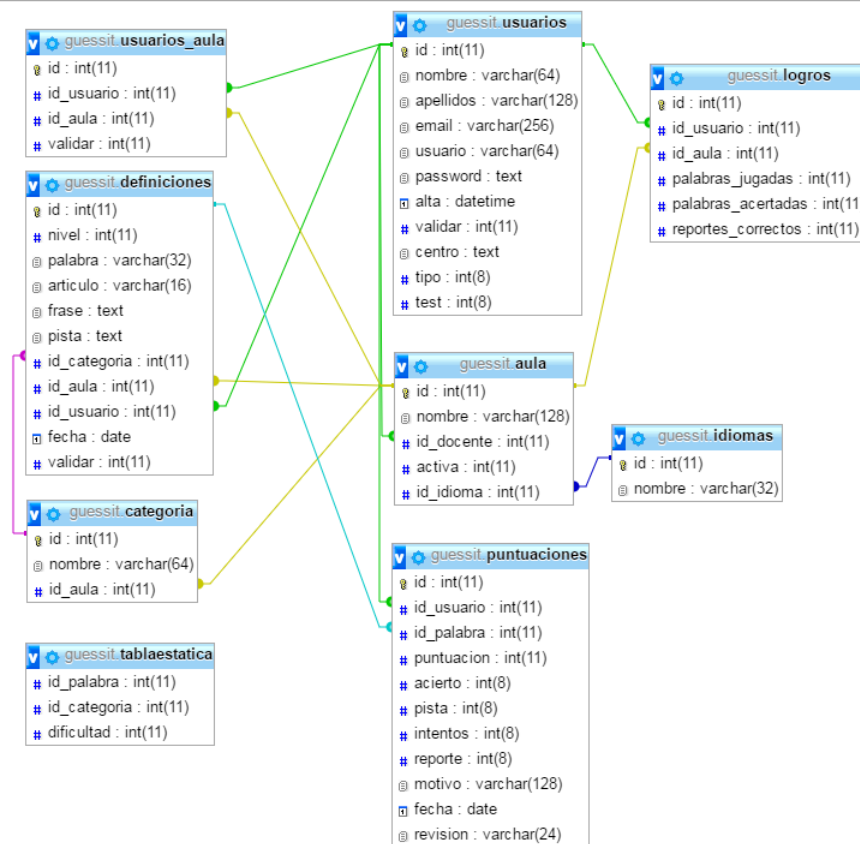


Figura 5.1: Diseño: Diagrama de relaciones de la base de datos.

Las tablas son las siguientes:

- **Usuarios:** tabla que almacena los datos de los usuarios del sistema. Cabe destacar los campos “tipo” y “test”, el primero se usa para diferenciar los tipos de usuarios del sistema (usuarios normales, docentes y administrador) y el segundo campo se usa para saber qué algoritmo de selección de definiciones usar con ese usuario.
- **Definiciones:** tabla que almacena el contenido del juego.
- **Aula:** es usada para almacenar la información de los grupos que se pueden crear en el sistema.
- **Usuarios_aula:** tabla que guarda la relación entre usuarios y grupos, aquí se administran las solicitudes pendientes para entrar a los grupos.
- **Categoría:** tabla que guarda todas las categorías creadas por los docentes en los grupos ya creados en el sistema.
- **Logros:** tabla que almacena el progreso de los usuarios.

- **Puntuaciones:** tabla donde se almacenan las partidas que van jugando los usuarios, junto con la evaluación o reporte que éstos les dan.
- **Idiomas:** tabla que almacena los idiomas, dados de alta por el administrador, disponibles a la hora de crear grupos.
- **tablaestatica:** tabla que se usa en el algoritmo de selección de definiciones de Alberto Cejas Sánchez.

Aquí el script para crear las tablas:

```
1  -- phpMyAdmin SQL Dump
2  -- version 4.0.10deb1
3  -- http://www.phpmyadmin.net
4  --
5  -- Servidor: localhost
6  -- Tiempo de generacion: 02-09-2016 a las 01:06:43
7  -- Version del servidor: 5.5.47-0ubuntu0.14.04.1
8  -- Version de PHP: 5.5.9-1ubuntu4.14
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
   */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
   */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
   */;
17 /*!40101 SET NAMES utf8 */;
18
19 --
20 -- Base de datos: `guessit`
21 --
22 CREATE DATABASE IF NOT EXISTS `guessit` DEFAULT CHARACTER SET
   utf8 COLLATE utf8_general_ci;
23 USE `guessit`;
24
25 DELIMITER $$
26 --
27 -- Procedimientos
28 --
29 DROP PROCEDURE IF EXISTS `debug_msg`$$
30 CREATE DEFINER='root'@'localhost' PROCEDURE `debug_msg`(enabled
   INTEGER, msg VARCHAR(255))
31 BEGIN
32     IF enabled THEN BEGIN
33         select concat("** ", msg) AS '** DEBUG:';
34     END; END IF;
35 END$$
36
37 DROP PROCEDURE IF EXISTS `proc_generar_tabla`$$
```

```
38 CREATE DEFINER='root'@'localhost' PROCEDURE `prc_generar_tabla
    `()
39 BEGIN
40
41     DECLARE id_palabra_actual INT;
42     DECLARE puntuacion_final FLOAT;
43     DECLARE puntuacion_antigua FLOAT;
44     DECLARE puntuacion_nueva FLOAT;
45     DECLARE intervalo INT DEFAULT 15;
46     DECLARE no_more_rows INT DEFAULT 0;
47     DECLARE done BOOLEAN DEFAULT 0;
48     DECLARE var1 INT;
49     DECLARE var2 INT;
50     DECLARE var3 INT;
51
52
53
54     -- Obtenemos las medias antes de X dias y despues
55     DECLARE medias_cursor CURSOR FOR
56     SELECT definiciones.id, viejas, nuevas
57     FROM definiciones LEFT JOIN
58     (
59         SELECT id_palabra, AVG(puntuacion)-1 AS nuevas
60         FROM puntuaciones
61         WHERE fecha > DATE_SUB(CURDATE(), INTERVAL
            intervalo DAY)
62         GROUP BY id_palabra
63     ) AS t1 ON definiciones.id = t1.id_palabra
64     LEFT JOIN
65     (
66         SELECT id_palabra, AVG(puntuacion)-1 as viejas
67         FROM puntuaciones
68         WHERE fecha <= DATE_SUB(CURDATE(), INTERVAL
            intervalo DAY)
69         GROUP BY id_palabra
70     ) AS t2 ON definiciones.id = t2.id_palabra
71     GROUP BY definiciones.id;
72     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
73
74     -- Borra la tabla tabla que contiene la las bolas que
75     -- representan
76     DROP TABLE IF EXISTS tablaestatica;
77
78     CREATE TABLE tablaestatica
79     (
80         id_palabra INT NOT NULL,
81         id_categoria INT NOT NULL,
82         dificultad INT NOT NULL
83     );
84
85
86     -- Recorre la seleccion y establece antigua*0.4 + nueva
87     *0.6
```



```
87      -- Teniendo en cuenta casos especiales donde no se haya
88      jugado en
89      -- dicho periodo
90      OPEN medias_cursor;
91
92      -- Comienza el bucle
93      REPEAT
94
95          FETCH medias_cursor INTO id_palabra_actual,
96          puntuacion_antigua, puntuacion_nueva;
97
98          /*-- Si no hay resultados sal de aqui
99          IF no_more_rows THEN
100              CLOSE medias_cursor;
101              LEAVE actualiza_medias;
102          END IF;*/
103
104          -- Calcula la puntuacion final
105          IF puntuacion_antigua IS NULL AND
106          puntuacion_nueva IS NULL THEN
107              SET puntuacion_final = 3;
108          ELSEIF puntuacion_antigua IS NULL THEN
109              SET puntuacion_final = puntuacion_nueva
110              ;
111          ELSEIF puntuacion_nueva IS NULL THEN
112              SET puntuacion_final =
113              puntuacion_antigua;
114          ELSE
115              SET puntuacion_final =
116              puntuacion_antigua*0.4 +
117              puntuacion_nueva*0.6;
118          END IF;
119
120          -- Para hacer mas representativa la diferencia
121          dada por los decimales,
122          -- multiplicamos la puntuacion final por 8 (por
123          ejemplo)
124          SET puntuacion_final = ROUND(puntuacion_final *
125          8);
126
127          -- Cache the data to store
128          SELECT definiciones.id, definiciones.
129          id_categoria, definiciones.nivel
130          INTO var1, var2, var3
131          FROM definiciones
132          WHERE id = id_palabra_actual;
133
134          -- Actualiza la tabla de bolas segun la
135          informacion anterior
136          WHILE puntuacion_final > 0 DO
137              -- Insertamos la fila tantas veces como el
138              redondeo de
139              -- puntuacion_final indique
140              INSERT INTO tablaestatica(id_palabra,
```

```
128         id_categoria, dificultad)
129         VALUES(var1,var2,var3);
130         SET puntuacion_final = puntuacion_final
131           - 1;
132     END WHILE;
133
134     -- Fin del bucle
135     UNTIL done END REPEAT;
136
137     CLOSE medias_cursor;
138
139 END$$
140
141 DELIMITER ;
142
143 -----
144
145 --
146 -- Estructura de tabla para la tabla `aula`
147 --
148 DROP TABLE IF EXISTS `aula`;
149 CREATE TABLE IF NOT EXISTS `aula` (
150   `id` int(11) NOT NULL AUTO_INCREMENT,
151   `nombre` varchar(128) NOT NULL,
152   `id_docente` int(11) NOT NULL,
153   `activa` int(11) NOT NULL,
154   `id_idioma` int(11) NOT NULL,
155   PRIMARY KEY (`id`)
156 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=15 ;
157
158 -----
159
160 --
161 -- Estructura de tabla para la tabla `categoria`
162 --
163
164 DROP TABLE IF EXISTS `categoria`;
165 CREATE TABLE IF NOT EXISTS `categoria` (
166   `id` int(11) NOT NULL AUTO_INCREMENT,
167   `nombre` varchar(64) NOT NULL,
168   `id_aula` int(11) NOT NULL,
169   PRIMARY KEY (`id`)
170 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=53 ;
171
172 -----
173
174 --
175 -- Estructura de tabla para la tabla `definiciones`
176 --
177
178 DROP TABLE IF EXISTS `definiciones`;
179 CREATE TABLE IF NOT EXISTS `definiciones` (
```

```
180 `id` int(11) NOT NULL AUTO_INCREMENT,
181 `nivel` int(11) NOT NULL,
182 `palabra` varchar(32) NOT NULL,
183 `articulo` varchar(16) DEFAULT NULL,
184 `frase` text NOT NULL,
185 `pista` text NOT NULL,
186 `id_categoria` int(11) NOT NULL,
187 `id_aula` int(11) NOT NULL,
188 `id_usuario` int(11) NOT NULL,
189 `fecha` date NOT NULL,
190 `validar` int(11) NOT NULL,
191 PRIMARY KEY (`id`)
192 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2359 ;
193
194 -- -----
195
196 --
197 -- Estructura de tabla para la tabla `idiomas`
198 --
199
200 DROP TABLE IF EXISTS `idiomas`;
201 CREATE TABLE IF NOT EXISTS `idiomas` (
202   `id` int(11) NOT NULL AUTO_INCREMENT,
203   `nombre` varchar(32) NOT NULL,
204   PRIMARY KEY (`id`)
205 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=8 ;
206
207 -- -----
208
209 --
210 -- Estructura de tabla para la tabla `logros`
211 --
212
213 DROP TABLE IF EXISTS `logros`;
214 CREATE TABLE IF NOT EXISTS `logros` (
215   `id` int(11) NOT NULL,
216   `id_usuario` int(11) NOT NULL,
217   `id_aula` int(11) NOT NULL,
218   `palabras_jugadas` int(11) NOT NULL,
219   `palabras_acertadas` int(11) NOT NULL,
220   `reportes_correctos` int(11) NOT NULL,
221   PRIMARY KEY (`id`),
222   KEY `id` (`id`)
223 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
224
225 -- -----
226
227 --
228 -- Estructura de tabla para la tabla `puntuaciones`
229 --
230
231 DROP TABLE IF EXISTS `puntuaciones`;
232 CREATE TABLE IF NOT EXISTS `puntuaciones` (
233   `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
234     'id_usuario' int(11) NOT NULL,
235     'id_palabra' int(11) NOT NULL,
236     'puntuacion' int(11) NOT NULL,
237     'acierto' int(8) NOT NULL,
238     'pista' int(8) NOT NULL,
239     'intentos' int(8) NOT NULL,
240     'reporte' int(8) NOT NULL,
241     'motivo' varchar(128) DEFAULT NULL,
242     'fecha' date NOT NULL,
243     'revision' varchar(24) NOT NULL DEFAULT 'no',
244     PRIMARY KEY ('id')
245 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=142625 ;
246
247 -----
248
249 --
250 -- Estructura de tabla para la tabla 'tablaestatica'
251 --
252
253 DROP TABLE IF EXISTS 'tablaestatica';
254 CREATE TABLE IF NOT EXISTS 'tablaestatica' (
255     'id_palabra' int(11) NOT NULL,
256     'id_categoria' int(11) NOT NULL,
257     'dificultad' int(11) NOT NULL
258 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
259
260 -----
261
262 --
263 -- Estructura de tabla para la tabla 'usuarios'
264 --
265
266 DROP TABLE IF EXISTS 'usuarios';
267 CREATE TABLE IF NOT EXISTS 'usuarios' (
268     'id' int(11) NOT NULL AUTO_INCREMENT,
269     'nombre' varchar(64) NOT NULL,
270     'apellidos' varchar(128) NOT NULL,
271     'email' varchar(256) NOT NULL,
272     'usuario' varchar(64) DEFAULT NULL,
273     'password' text NOT NULL,
274     'alta' datetime NOT NULL,
275     'validar' int(11) NOT NULL,
276     'centro' text,
277     'tipo' int(8) NOT NULL COMMENT '0 = alumno, 1 = docente',
278     'test' int(8) NOT NULL,
279     PRIMARY KEY ('id')
280 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=123 ;
281
282 -----
283
284 --
285 -- Estructura de tabla para la tabla 'usuarios_aula'
286 --
287
```

```
288 DROP TABLE IF EXISTS `usuarios_aula`;
289 CREATE TABLE IF NOT EXISTS `usuarios_aula` (
290   `id` int(11) NOT NULL AUTO_INCREMENT,
291   `id_usuario` int(11) NOT NULL,
292   `id_aula` int(11) NOT NULL,
293   `validar` int(11) NOT NULL,
294   PRIMARY KEY (`id`)
295 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=226 ;
296
297 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
298 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
299   */;
300 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

5.3. Diseño de clases

A continuación, se muestra una lista con las diferentes clases Java necesarias para la realización del cliente. Junto a cada una de las clases habrá una pequeña descripción sobre la labor que desempeña cada una:

MainGame Clase principal de LibGDX que maneja lo necesario para iniciar el juego, en este caso, la aplicación cliente de nuestro sistema.

MainScreen Screen de LibGDX que contiene el menú inicial del cliente junto al logo de la aplicación.

LoginScreen Screen de LibGDX que muestra los campos de texto a rellenar con la información de login del usuario.

RegisterScreen Screen de LibGDX que muestra los campos de texto a rellenar con la información de registro del usuario.

CreditsScreen Screen de LibGDX que muestra a los participantes del proyecto y su rol.

NoteBookScreen Screen de LibGDX que muestra las palabras guardadas en el bloc de notas, además de un cuadro de texto para cada una para anotar dudas.

UserGroupsScreen Screen de LibGDX que muestra un botón para abrir la ventana de solicitudes a grupos y los grupos a los que ya tiene acceso el usuario.

JoinGroupsScreen Screen de LibGDX que muestra los grupos disponibles en el sistema, que no tenga el usuario ya validado, para enviar solicitudes de admisión en ellos.

MenuGameScreen Screen de LibGDX que muestra el menú de acciones disponibles dentro del grupo.

ConfigGameScreen Screen de LibGDX que muestra los elementos necesarios para configurar una ronda de juego.

StatsMenuScreen Screen de LibGDX que muestra un submenú con las opciones de estadística que desee ver el usuario.

StatsScreen Screen de LibGDX que muestra el progreso en el grupo del usuario, así como las aportaciones de éste que han sido reportadas.

GoalsScreen Screen de LibGDX que muestra los logros obtenidos en el grupo.

NewDefScreen Screen de LibGDX que muestra un formulario con una palabra a añadir una nueva definición en el sistema.

GameScreen Screen de LibGDX y ventanas menores que muestran los elementos necesarios para jugar.

Definition Clase de mapeo, para facilitar las operaciones en las Screens y peticiones HTTP.

GEngine Pequeño “engine” de comparación de cadenas de caracteres y procesamiento de cadenas para ocultar la palabra a buscar, además de proveer diferentes pigmentaciones a las respuestas y, en el caso de que lo tenga la definición, a los artículos.

Group Clase de mapeo, para facilitar su uso en las Screens y peticiones HTTP.

Strings_I18N Clase auxiliar para la transformación del idioma de la interfaz dependiendo del idioma de trabajo del grupo.

UserInfo Clase de mapeo, para almacenar la información del usuario de forma local.

Utils Clase auxiliar que carga los recursos visuales y texturas al arrancar la aplicación.

El resto de clases son parte del framework LibGDX, incluidas en el paquete Scene2D y Scene2D.ui.

5.4. Interfaz gráfica

Tras los resultado que se han obtenido en la fase de análisis del sistema, es necesario desarrollar una interfaz sencilla y agradable para el usuario de la misma. En el diseño de estos, se intentará en todo momento que el flujo de ejecución sea lo más intuitivo posible.

En el siguiente diagrama podemos ver las distintas opciones de navegación dentro del juego:

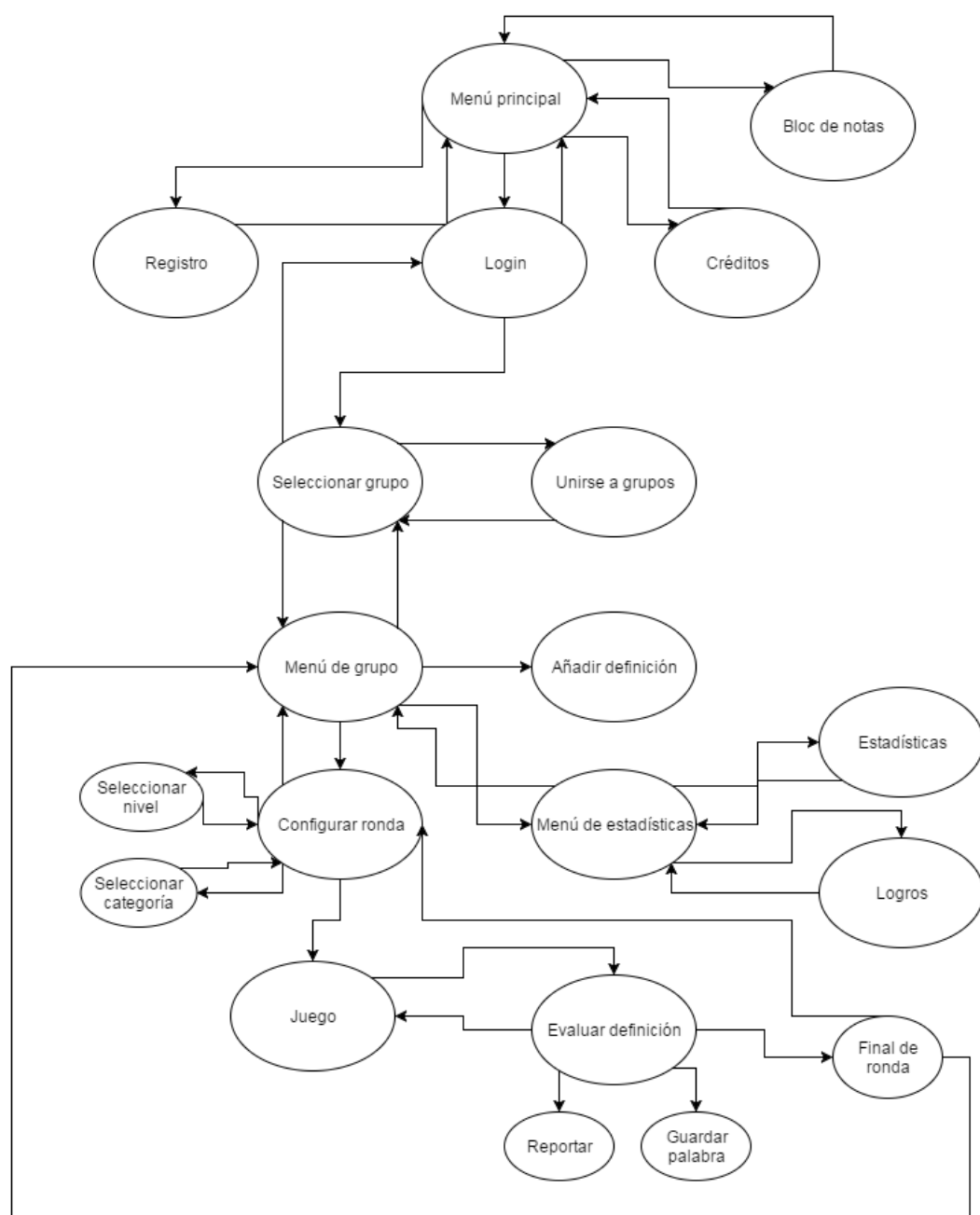


Figura 5.2: Diseño: Diagrama de interacción entre pantallas

Y aquí el resultado del desarrollo de las pantallas:

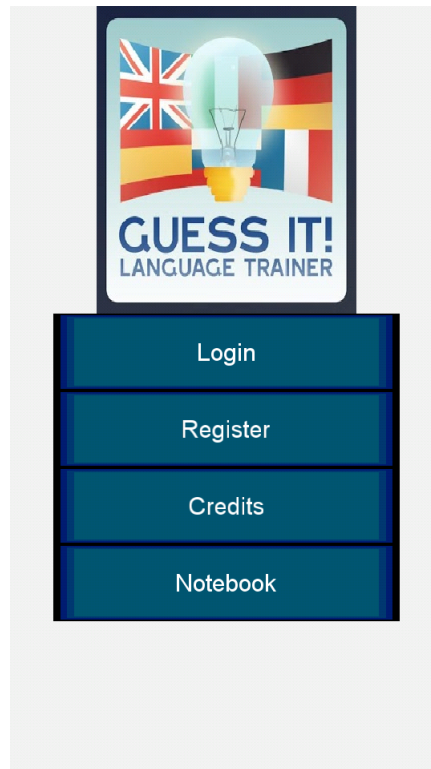
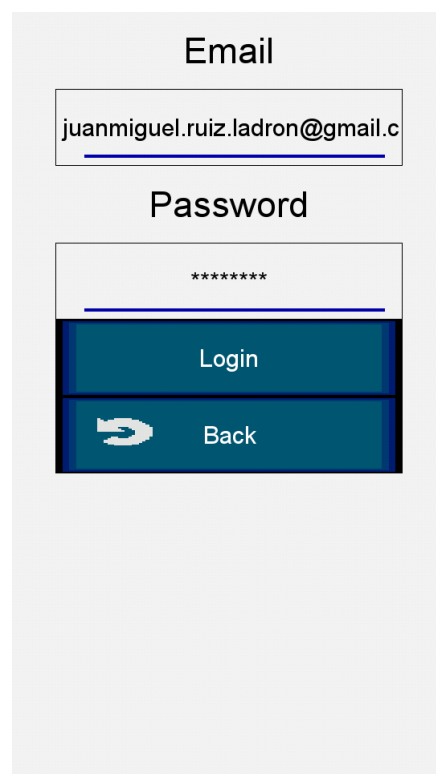


Figura 5.3: Diseño: Pantalla principal



The image shows a login screen design on a light gray background. At the top, the word "Email" is centered in a black sans-serif font. Below it is a white rectangular input field containing the text "juanmiguel.ruiz.ladron@gmail.c". A blue horizontal line is positioned at the bottom of this field. Below the email field, the word "Password" is centered in a black sans-serif font. Below it is a white rectangular input field containing seven asterisks "*****". A blue horizontal line is positioned at the bottom of this field. Below the password field are two dark blue rectangular buttons stacked vertically. The top button contains the word "Login" in white sans-serif font. The bottom button contains a white left-pointing arrow icon followed by the word "Back" in white sans-serif font.

Figura 5.4: Diseño: Pantalla de login

The image shows a mobile application registration screen. It has a light gray background. At the top, there are two input fields: 'Name' and 'Last name'. Below them is a single 'Email' input field. Further down are 'Username' and 'Password' input fields. At the bottom of the form area are two dark blue buttons: 'Register' and 'Back'. Below the buttons is a virtual keyboard with a light gray background and dark gray text. The keyboard has three rows of letters, a fourth row with a shift key, a spacebar, and a delete key, and a fifth row with a language/emoji key, a spacebar, and a return key.

Figura 5.5: Diseño: Pantalla de registro

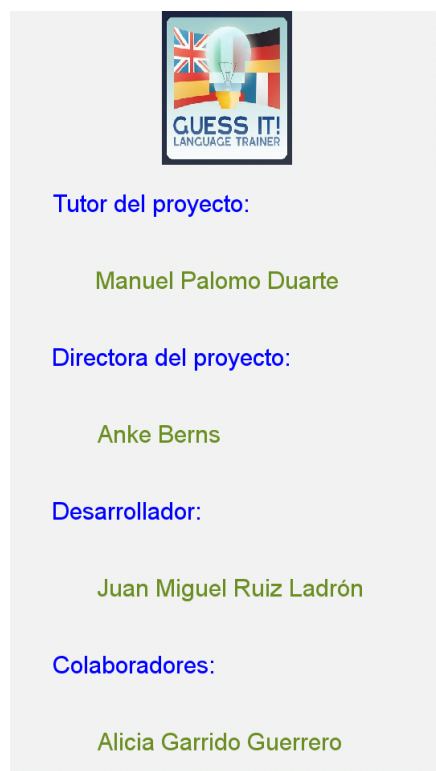


Figura 5.6: Diseño: Primera parte de la pantalla de créditos

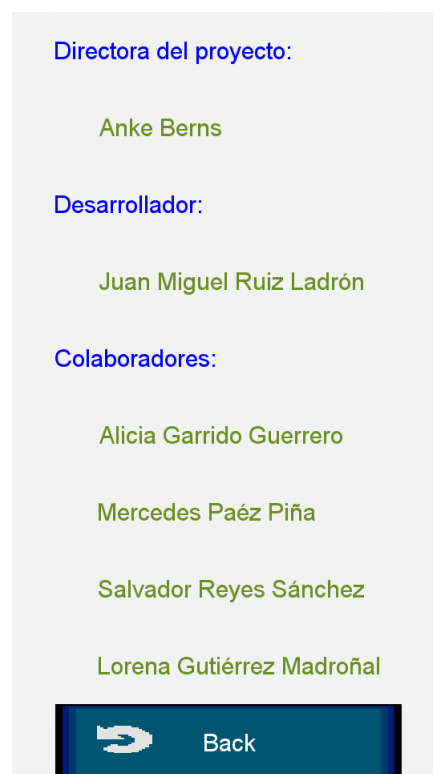


Figura 5.7: Diseño: Segunda parte de la pantalla de créditos

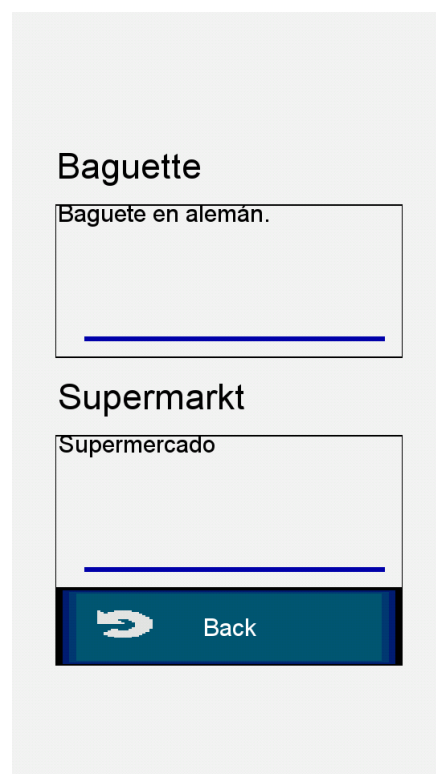


Figura 5.8: Diseño: Pantalla del bloc de notas

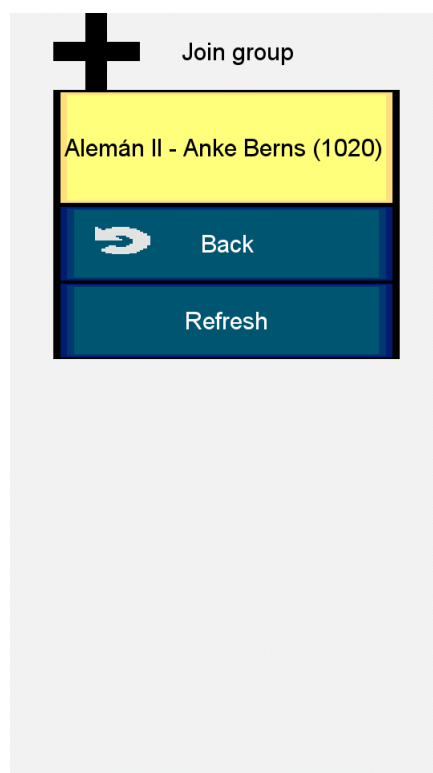


Figura 5.9: Diseño: Pantalla de selección de grupo

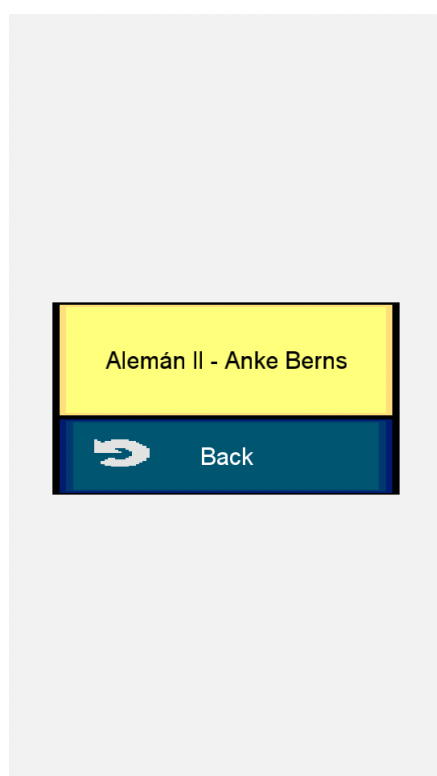


Figura 5.10: Diseño: Pantalla de envío de solicitudes a grupos



Figura 5.11: Diseño: Pantalla de opciones del grupo



Figura 5.12: Diseño: Pantalla de configuración de una ronda

Wähl die Kategorie:

Essen (43)
Wetter (8)
Wochentage (6)
Freizeit (20)
Aussehen (19)

Figura 5.13: Diseño: Pantalla de selección de categorías

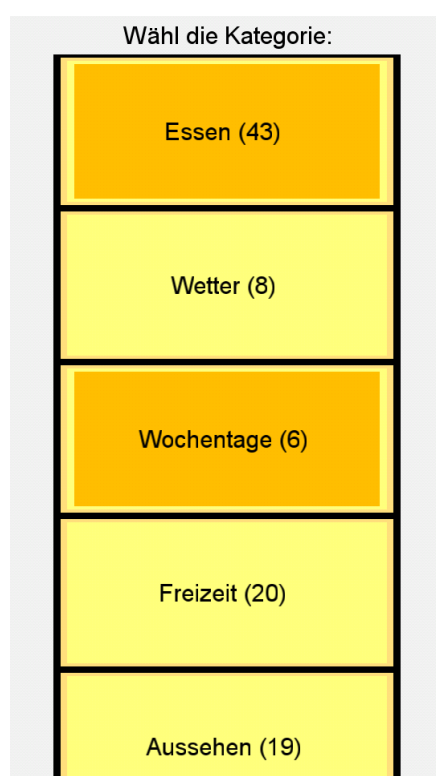


Figura 5.14: Diseño: Pantalla de selección de categorías con algunas seleccionadas

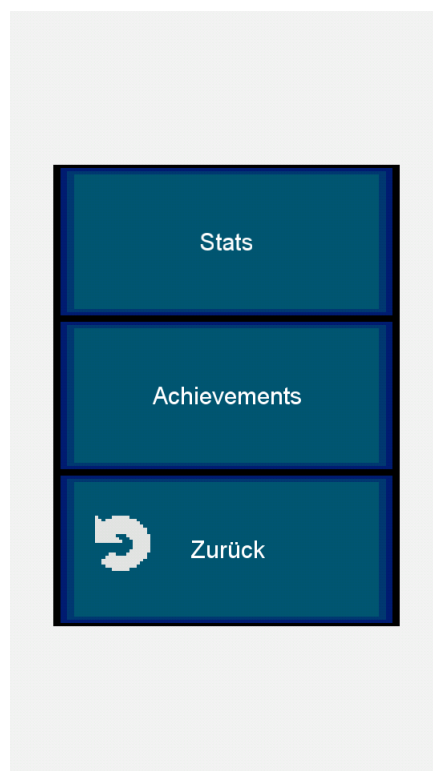


Figura 5.15: Diseño: Pantalla de menú de estadísticas

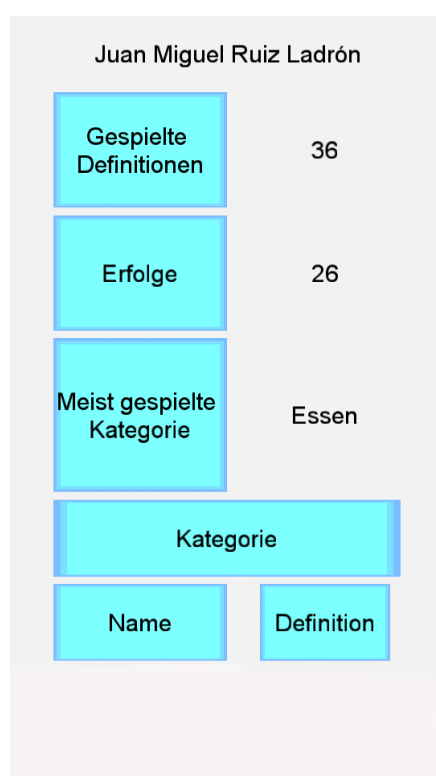
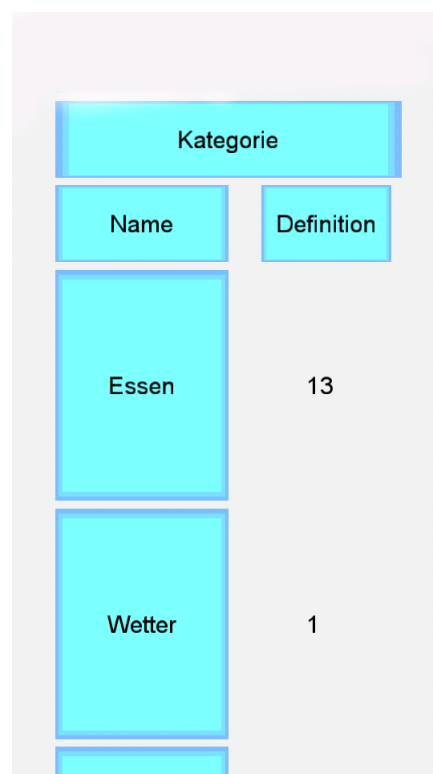


Figura 5.16: Diseño: Primera parte de pantalla de estadísticas



Kategorie	
Name	Definition
Essen	13
Wetter	1

Figura 5.17: Diseño: Segunda parte de pantalla de estadísticas

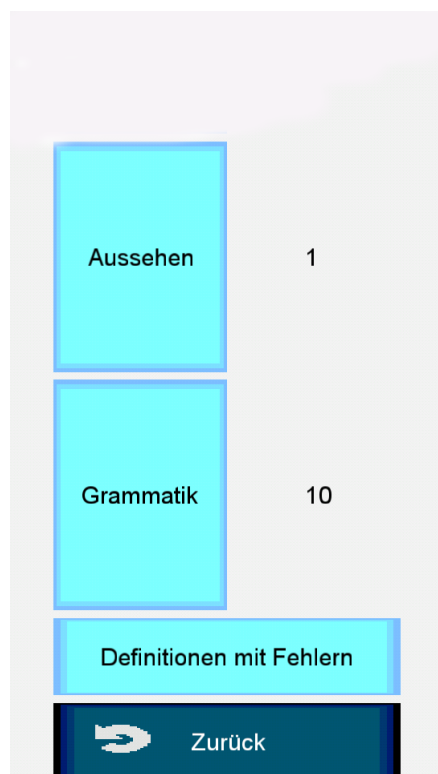


Figura 5.18: Diseño: Tercera parte de pantalla de estadísticas


Category: **Wohnung**

Article: **das** Level: **1**

Word: **Bett**

Enter the sentence

Enter the hint

 **Senden**


 **Zurück**

Figura 5.19: Diseño: Pantalla de añadir una nueva definición

Auf Englisch sagt man
"supermarket". Und auf
Deutsch? *****.

Antwort:

der

Rate! Tipp

Figura 5.20: Diseño: Pantalla de juego

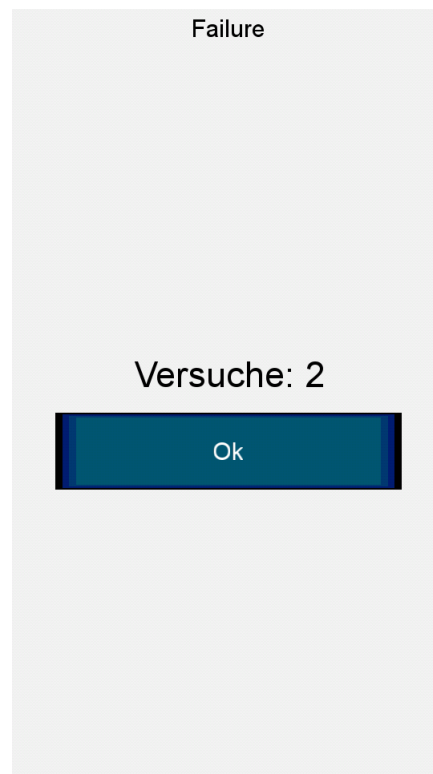


Figura 5.21: Diseño: Pantalla de número de intentos

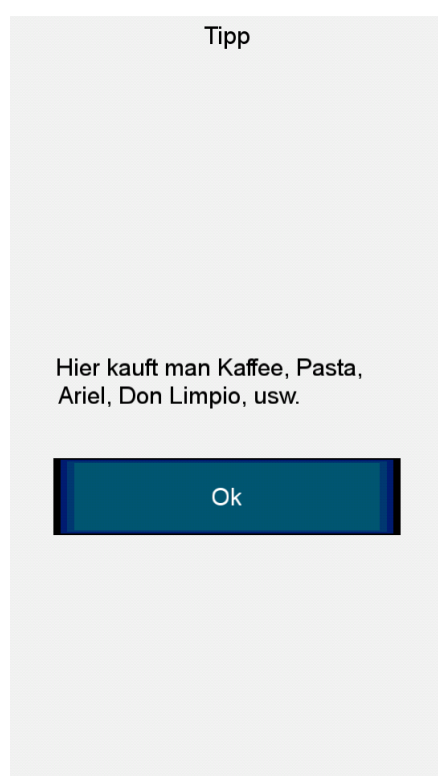




Figura 5.22: Diseño: Pantalla de pista

Evaluier die Definition!

Auf Englisch sagt man
"supermarket". Und auf
Deutsch? **Supermarkt**

* Sind die
Substantive groß
geschrieben? 

☒ Ja ☐ Nein

 Report



 Senden	 Speichern
--	---

Figura 5.23: Diseño: Pantalla de evaluación de la definición

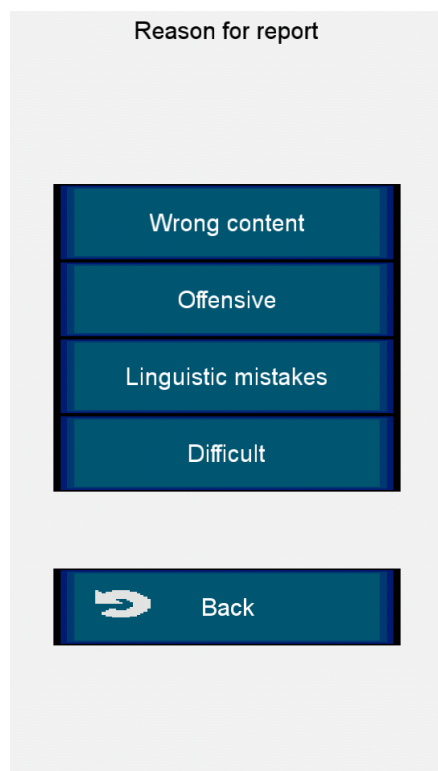


Figura 5.24: Diseño: Pantalla de motivo de reporte

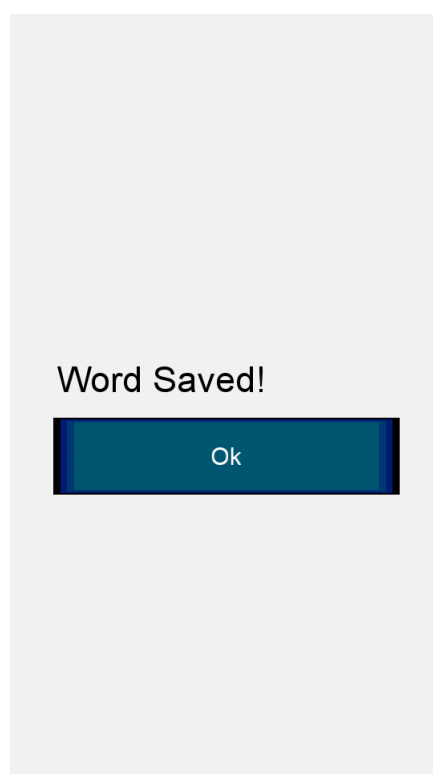


Figura 5.25: Diseño: Pantalla de confirmación de palabra guardada en el bloc

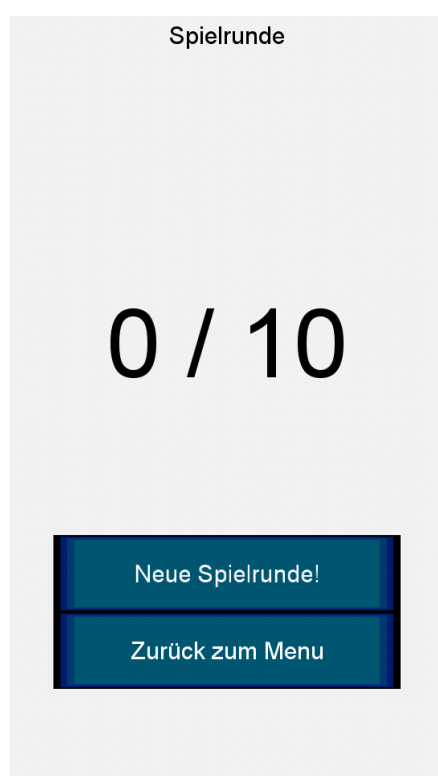


Figura 5.26: Diseño: Pantalla de fin de ronda

Implementación

En ésta sección trataremos aquellos problemas que fueron surgiendo en el desarrollo del proyecto, que se tuvieron que ir resolviendo para un desarrollo continuo.

Por tanto, explicaremos junto a los problemas, su solución.

A continuación se explicarán aspectos que no han sido explicados con anterioridad en los apartados previos a éste.

6.1. Layout y resolución

Cualquier dispositivo con un sistema Android superior a la versión 2.3 puede ejecutar el cliente. Debido a que cada dispositivo tiene unas configuraciones de pantalla distintas, distintas resoluciones, profundidad de pantalla, etc. LibGDX actúa en consecuencia, el framework crea un entorno OpenGL internamente, por lo que ciertos parámetros, como los de la pantalla, son adaptados al entorno.

Los elementos visuales funcionan con una constante multiplicativa para mantener la misma relación de aspecto en cada dispositivo.

6.2. Internacionalización

Otro problema fue el idioma de la interfaz, en la fase de diseño se planteó que el idioma de la interfaz dependa del idioma del grupo, haciendo así que se use el Inglés como idioma común antes de la selección del grupo en el que jugar. El idioma de la interfaz cambia a la configuración que tenga el grupo.

6.3. Multiplataforma

Un problema que ocurría con frecuencia en el anterior proyecto era la compatibilidad, el anterior cliente sólo funcionaba en Android, haciendo imposible que usuarios de Iphone pudieran participar sin usar un emulador (con escaso rendimiento).

Este problema se solucionó sacando una versión de Java del cliente Android, algo que LibGDX realiza de forma automática. Así pues, usuarios de Linux, Windows y Mac OSX podían disfrutar del juego de forma nativa.

6.4. Creación de grupos

La primera característica es la creación de grupos separados, con contenido que puede ser diferente entre ellos, para llevar un control absoluto de él, de forma que un grupo de alumnos no se mezcle con gente externa.

En la creación de los grupos se incluyen las categorías a jugar en él, además de ser el enlace con la subida de definiciones a la plataforma.

6.5. Subida de contenido a los grupos

Los docentes disponen de dos métodos para añadir contenido a sus grupos, el primero de ellos, un formulario muy sencillo para subir una sola definición cada vez. El segundo método, la subida de un archivo CSV con un formato concreto, que se almacenará de forma automática en la base de datos.

6.6. Selección de definiciones

Se han puesto a prueba dos algoritmos diferentes para la selección de definiciones.

El primero de ellos es el algoritmo usado por Alberto Cejas Sánchez en su proyecto “GuessIt! Language Trainer”, explicado en su memoria.

El segundo algoritmo ha sido diseñado para contrastar datos con el primer algoritmo ya mencionado. Funciona de la siguiente forma:

- Se divide cada partida en rondas, siendo de 1 a N rondas. Y cada ronda con X definiciones (en ésta ocasión se escogieron 10 definiciones por ronda).
- Se escoge la mitad de X como número de definiciones a las que aplicar el algoritmo, la otra mitad serán, mientras queden, definiciones nunca jugadas por el usuario.
- Se obtienen dos porcentajes, uno de definiciones jugadas acertadas frente a las totales.
- De la mitad de las definiciones que serán procesadas por el algoritmo, se hace uso del porcentaje de acertadas y se aplica a la mitad de X seleccionada, dando una cantidad “A”.

- Restamos de $X/2$ la cantidad “A” y obtenemos la cantidad “B”.
- “A” nos marca la cantidad de definiciones **nuevas** que han de escogerse.
- “B” nos indica la cantidad de definiciones **ya jugadas** que han de escogerse.
- Componemos la ronda con las definiciones obtenidas de “A”, las de “B” y las nuevas de la otra mitad de X. Éstas se envían para jugarse.

6.7. Generador de estadísticas

A través de un sencillo formulario, el docente puede configurar ciertos parámetros como, qué usuarios, qué tipo de estadística quiere, en qué rango de fechas y el nivel o los niveles que desea visualizar de su grupo.

La salida es dada por una gráfica interactiva, aunque de información estática, generada a partir de Chart.js. Aquí vemos algunos ejemplos:

Figura 6.1: Implementación: Generador de estadísticas

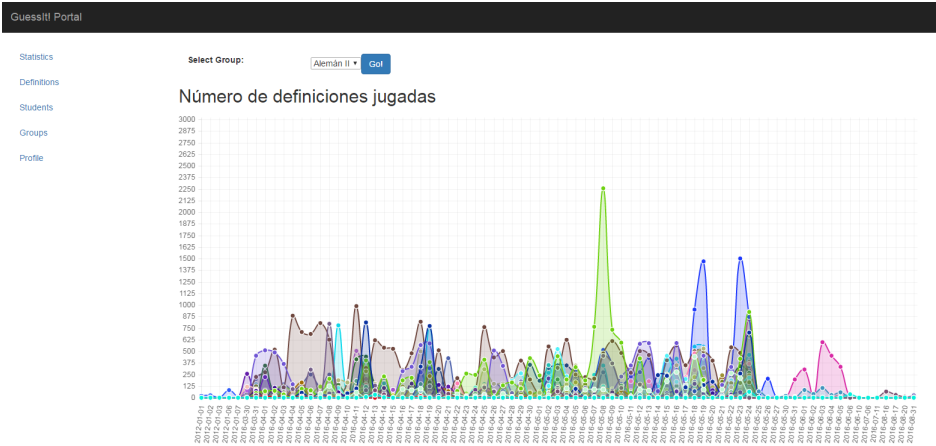


Figura 6.2: Implementación: Definiciones jugadas

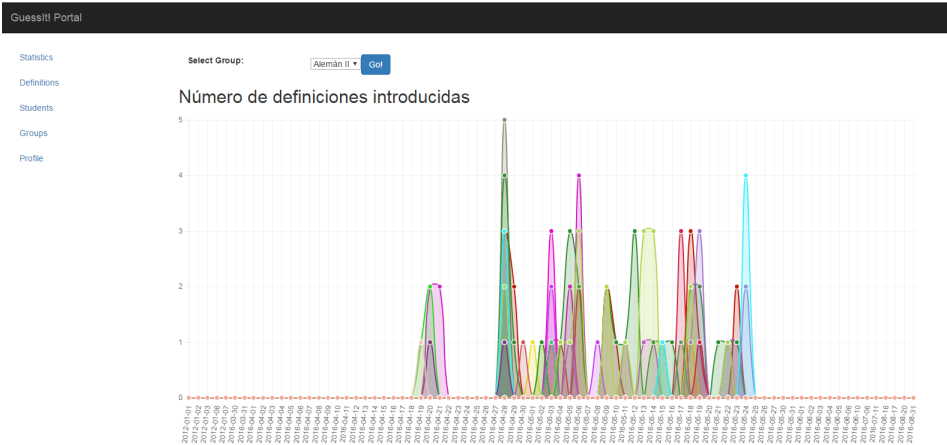


Figura 6.3: Implementación: Definiciones introducidas

Pruebas y validaciones

Las pruebas en éste tipo de sistemas son exhaustivas y deben ser determinantes para mejorar el rendimiento y, en efecto, la experiencia de los usuarios.

Cada módulo se ha probado individualmente tanto de forma local, como por colaboradores del proyecto, aportando un feedback muy importante y determinante en la mayoría de los casos.

Además, se realizaron pruebas de integridad con otros módulos, como por ejemplo, la conexión a la base de datos remota, haciendo que el proyecto saliese del ámbito local.

En definitiva, la organización de los casos de pruebas fue la siguiente:

- Tras finalizar la implementación de cada módulo, se realizaban pruebas unitarias sobre estos.
- A medida que distintos módulos que anteriormente probados individualmente, debían colaborar entre ellos, se llevaban a cabo pruebas de integración.
- Con las distintas versiones jugables se realizaban pruebas de jugabilidad.
- Pruebas de interfaz.

7.1. Pruebas unitarias

Esta pruebas se realizaron junto a la fase de implementación, conforme se implementaban nuevos módulos necesarios para la aplicación se realizaban pruebas individuales sobre estos módulos. De esta forma se buscaban todos los caminos posibles que podría dar cada módulo, teniendo en cuenta aquellos que fueran más predispuesto a fallos.

De esta forma todas las sentencias se ejecutaban como mínimo una vez y los posibles fallos se encontraban de una forma más sencilla. Por lo que también era más fácil localizar donde estaba el problema y afrontar la solución de éste.

Éstas pruebas se ejecutaron en el IDE usado en el proyecto, IntelliJ.

7.2. Pruebas de integración

Conforme aparecían nuevos módulos, cuya implementación era necesaria y a su vez estos requerían el uso de otro módulos que posteriormente habían sido probados individualmente, se realizaban pruebas de integración entre dichos módulos.

A medida que se avanzaba en el desarrollo del sistema, eran precisas pruebas de mayor envergadura, realizando dichas pruebas en el conjunto completo del sistema, el cliente Android, el servidor y la interfaz web.

Éstas pruebas se realizaron junto a los colaboradores del proyecto, elaborando un completo feedback con cada nuevo error que se producía en el sistema al incluir una nueva funcionalidad.

Por ejemplo, en el caso de los algoritmos de selección, a veces se introducía una palabra de una categoría que no había sido seleccionada. Problema que fue resuelto modificando las consultas que emplean los algoritmos.

7.3. Pruebas de jugabilidad

A lo largo del desarrollo, cuando la interfaz de usuario estuvo definida, se requerían pruebas para comprobar si la disposición y despliegue de los botones, campos de texto, colores y demás aspectos eran satisfactorios para el uso de la aplicación móvil.

Se realizó un exhaustivo feedback de dicho apartado, ya que los usuarios finales iban a ser alumnos, se necesitaba una armonía entre la funcionalidad y la apariencia.

Un problema que ocurría con frecuencia era la elección de colores en la aplicación Android. Se definieron una serie de colores, que contrastaron bien con el fondo, para los botones, campos de texto y solución de las definiciones.

7.4. Pruebas de usabilidad

Además de estudiar si era usable, se realizaron pruebas para determinar si la interfaz era agradable de manejar. La interacción entre los diferentes menús y los elementos en pantalla no debe contener fallos ni generar errores en los datos.

Conclusiones

Comentaremos en ésta sección las conclusiones finales en el desarrollo de éste sistema.

8.1. Objetivos

Los objetivos planteados en el proyecto: planificación, desarrollo acorde con lo documentado; se ha ido cumpliendo, en mayor medida.

Se consiguió terminar una versión totalmente funcional y probada el día de la presentación al aula que usaría el software.

Además de conseguir la primera premisa para elaborar éste proyecto, solventar todos los problemas de compatibilidad con los diferentes dispositivos, tanto portátiles como sobremesa.

En el día de su primer uso real fuera de un entorno controlado salieron algunos problemas que se solucionaron en la misma sesión, sin que se tuviese que parar el uso del cliente o del servidor para solventarlos.

Por tanto, concluyo que se han cumplido con los objetivos propuestos para éste proyecto.

8.2. Conclusiones personales

Aún conociendo el framework con el que está implementado éste proyecto, y conociendo algunas de las tecnologías web usadas anteriormente, éste proyecto ha supuesto un reto para mí.

Nunca antes había enfrentado un desarrollo completo de un proyecto de ésta envergadura, con la llegada del feedback a cada versión se avecinaban días y días de trabajo,

además de, paralelamente, sacar nuevas versiones con más funcionalidades a probar.

Agradezco la experiencia obtenida al enfrentarme a la presión de las fechas para las pruebas y las fases del desarrollo del proyecto.

Echando la vista atrás, a cuando la realización de éste proyecto era una simple idea, rediseñar un proyecto de tal envergadura como GuessIt! parecía imposible. El proyecto era grande, y pude participar en su mantenimiento para alargar su vida un año más. Ahora, aún conservando su esencia, el proyecto ha conseguido llegar a tener una gran cantidad de características nuevas, además de solventar problemas de compatibilidad y, quién sabe, sacar el cliente para muchas más plataformas.

8.3. Mejoras y ampliaciones

Algunas ideas no pudieron ser desarrolladas por falta de tiempo y quedaron como ideas. En éste apartado se exponen las ideas que han quedado fuera del proyecto, pendientes de ser implementadas en un futuro:

- **Sinónimos:** Añadir soporte para sinónimos de modo que haya más palabras válidas para una misma definición.
- **Modo duelo:** Modo en el que retar a algún otro participante del grupo para conseguir puntos.
- **Puntuaciones y ranking:** Sistema de puntuaciones para obtener índices adicionales de esfuerzo en la aplicación. Además de poder ver un ranking con aquellos con mejor puntuación del grupo.
- **Versión HTML:** Con el fin de eliminar las instalaciones en dispositivos y tener un sistema 100 % multiplataforma, modificar el código fuente para que sea compatible con el entorno HTML5 y lanzar su versión responsiva web.
- **Incluir más logros:** Los logros que se encuentran disponibles en la plataforma son unos logros básicos que hacen referencia al uso de la aplicación.
- **Imágenes como ayuda:** Además de tener la mecánica de pistas, plantear y desarrollar que, a la vez que la pista, otorgar el acceso a una imagen referenciando a la palabra que adivinar.



Apéndice

Herramientas utilizadas

A.1. Lenguaje del cliente

El cliente se ha desarrollado con el framework LibGDX, disponible en el siguiente enlace:

<https://libgdx.badlogicgames.com/>

Está escrito y se usa en Java y permite exportar el proyecto en:

- Binario de Java ejecutable
- Apk de Android
- iOS
- HTML5
- Blackberry



Figura A.1: Apéndice A: LibGDX logo

A.2. Base de datos

Se decidió utilizar MYSQL por ser libre, sencilla y potente.



Figura A.2: Apéndice A: MySQL logo

A.3. Frontend

Se decidió utilizar PHP con Codeigniter como framework. El motivo principal es la facilidad de utilizar y aprender.



Figura A.3: Apéndice A: CodeIgniter logo

A.4. Backend

Se decidió utilizar PHP por su fiabilidad en entornos web.



Figura A.4: Apéndice A: PHP logo

A.5. Aseprite, Gimp e Inkscape

Programas de ilustración, retoque y creación de imágenes vectoriales totalmente OpenSource y de fácil aprendizaje.



Figura A.5: Apéndice A: Aseprite logo

A.6. Sistema de control de versiones

Para el control de versiones se ha usado Git, haciendo uso del entorno instalable en línea de comandos, además de tener instalados la versión de escritorio del entorno e integrar en IntelliJ el sistema de Git.



Figura A.6: Apéndice A: Git logo

A.7. Documentación del código

La documentación del código se ha llevado a cabo con **Javadoc**, herramienta de manejo sencillo que exporta un documento de buena legibilidad en formato HTML.

A.8. Redacción de la memoria

Para la completa realización de la memoria se ha usado \LaTeX . Es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.



Figura A.7: Apéndice A: \LaTeX logo

\LaTeX es un sistema de composición de textos que está formado mayoritariamente por órdenes (macros) construidas a partir de comandos de \TeX . \LaTeX es una herramienta práctica y útil pues, a su facilidad de uso, se une toda la potencia de \TeX .

A.9. Realización de diagramas: DIA

Para la realización de todos los diagramas necesarios que aparecen a lo largo de toda la memoria se ha usado el creador de diagramas Dia.

Dia es un programa de creación de diagramas en GNU/Linux, MacOS X, Unix y Windows, bajo la licencia GPL. Puede ser utilizado para dibujar diferentes tipos de diagramas. Actualmente cuenta con herramientas para dibujar diagramas entidad relación, diagramas UML, diagramas de flujo, diagramas de red, y muchos otros diagramas.



Figura A.8: Apéndice A: Dia logo



Apéndice

B.1. Manual de instalación

B.1.1. Cliente

El cliente se puede descargar desde el siguiente enlace:

<http://164.132.193.133/app/GuessIt-RELEASE.apk>

No dispone de versión de *Google Play* debido a la versión de GuessIt! de la cual no soy propietario.

Es necesario tener activada la opción “Fuentes desconocidas” del menú “Seguridad” del dispositivo Android para continuar la instalación.

También está disponible la versión Jar del cliente, para ejecutarlo en cualquier dispositivo de sobremesa:

http://164.132.193.133/jar/GuessIt_Windows_RELEASE.jar

Es necesario tener instalado el entorno de Java (JRE) en el equipo. Simplemente ejecuta el archivo JAR (ejecutable de Java).

B.1.2. Interfaz web

La interfaz web está disponible en el siguiente enlace:

<http://164.132.193.133/GuessIt/>

Se puede ver desde cualquier navegador e incluso desde cualquier dispositivo móvil.

Para su instalación es preciso tener instalado un entorno LAMP o el paquete para servidores Xampp. A continuación veremos la instalación en un servidor con Ubuntu instalado.

Servidor LAMP

El paquete LAMP no es más que un conjunto de software Open Source que conforman el software básico de un servidor.

Paso 1:

Lo primero es instalar Apache:

```
1 sudo apt-get update
2 sudo apt-get install apache2
```

Paso 2:

Instalar MySQL:

```
1 sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql
```

Y una vez tienes instalado todo lo anterior, debes iniciarlo:

```
1 sudo mysql_install_db
```

Cuando esté instalado, arranca el script con el siguiente comando:

```
1 sudo /usr/bin/mysql_secure_installation
```

Se te preguntará por una contraseña de root, se le facilita una y luego a todas las preguntas que aparecerán luego, se contesta que sí.

Paso 3:

Instalar PHP:

Abrimos una terminal y escribimos:

```
1 sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Es conveniente añadir php en el índice del directorio web:

```
1 sudo nano /etc/apache2/mods-enabled/dir.conf
```

Se debe añadir `index.php` al inicio de la cadena que aparecerá al configurar el archivo del índice del directorio:

```
1 <IfModule mod_dir.c>
2     DirectoryIndex index.php index.html index.cgi index.pl index.php index.xhtml
3     index.htm
4 </IfModule>
```

Paso 4:

Copiar directorio con la web:

Instalado el servidor con el conjunto LAMP, lo que resta es copiar la carpeta de “GuessItWeb” del repositorio de Git y copiarla en `/var/www/`.

Paso 5:

En la elaboración del proyecto se ha usado PHPMyAdmin para la gestión de la base de datos, en el proyecto se incluye el script de creación de las tablas de la base de datos. Sólo resta importar la base de datos con su estructura y modificar las direcciones de CodeIgniter y los accesos de los archivos php.



Apéndice

C.1. Manual de usuario

C.1.1. Manual del cliente

Sistema multiplataforma para aprendizaje de vocabulario contextualizado

Manual de usuario

Cliente Android

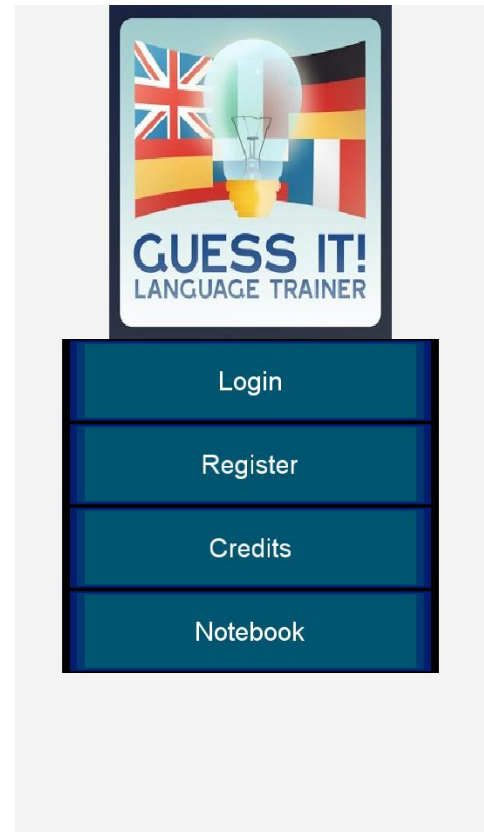
Instrucciones:

- Registro
- Unirse a grupo
- Entrar en grupo
- Configurar ronda
- Jugar
- Ver estadísticas

Registro

Lo primero es registrarnos en la plataforma para poder acceder a todas las características que ofrece.

Pulsamos sobre "Register" y aparecerán unos campos de texto que debemos rellenar para completar nuestro registro, pulsamos sobre "Register" y estaremos dentro de la plataforma.

The image shows the registration form within the app. It contains several input fields: 'Name' and 'Last name' at the top, followed by 'Email'. Below these are 'Username' and 'Password' fields. At the bottom of the form are two buttons: a dark blue 'Register' button and a dark blue button with a white back arrow icon labeled 'Back'. Below the form, a portion of an Android keyboard is visible, showing the letters 'q w e r t y u i o p' on the first row, 'a s d f g h j k l ñ' on the second row, and 'z x c v b n m' on the third row, along with a spacebar and a back arrow.

Unirse a grupo

Una vez registrados y logueados en la plataforma, ¡es hora de escoger un grupo!

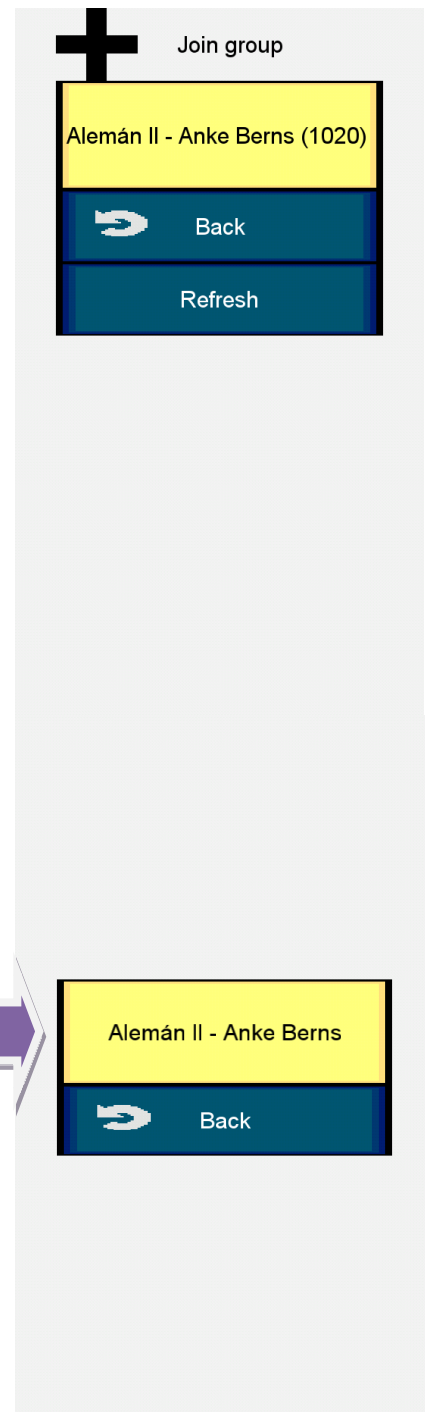
Si es nuestra primera vez, en la pantalla luego de loguearnos, sólo aparecerán tres botones: **"Join group"**, **"Refresh"** y **"Back"**

El botón amarillo con los datos de uno de los grupos no aparecen hasta que seamos validados como participantes.

Por tanto, pulsaremos sobre **"Join group"** y nos llevará a la siguiente ventana.

En ella aparecerán todos los grupos disponibles en el sistema, con el nombre del grupo y el docente que lo administra.

Pulsaremos sobre cada grupo al que queramos enviar la solicitud de admisión y esperaremos a que el docente nos acepte.



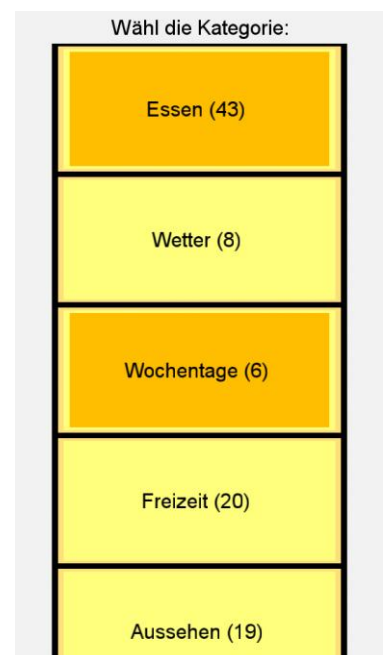
Entrar a un grupo

Una vez seamos aceptados en un grupo, nos aparecerá en la pantalla anterior a la de las solicitudes, entrando en uno cambiará el idioma de la interfaz a la que tenga configurada el grupo, como caso de ejemplo, el grupo de "**Alemán II**" está configurado con Alemán como idioma:



Configurar partida

Dando sobre la opción "Jugar", en el idioma en el que se encuentre la interfaz, entraremos en el menú para configurar una ronda de juego, pulsamos primero sobre el nivel a jugar, y luego sobre las categorías a jugar, al lado de cada categoría viene el número de definiciones del nivel que has seleccionado anteriormente.



Una vez acabemos la configuración, pulsamos sobre "Jugar" una vez más y ¡comenzará la partida!

Jugar

Nos aparecerá la definición cuya palabra a adivinar estará oculta, el artículo de la palabra, en caso de que lo tenga, y un cuadro de texto donde escribiremos la palabra que creamos correcta.

También está disponible una pista, que sólo podemos activarla si probamos a adivinar la palabra al menos una vez.

Auf Englisch sagt man "supermarket". Und auf Deutsch? *****.

Antwort:

der

Rate! Tipp


Al adivinar la palabra o agotar todos los intentos, aparecerá la pantalla para evaluar la definición, debemos responder a la pregunta que aparece ahí, o, si la definición tuviera fallos graves, reportarla.

Al reportar una definición, debemos seleccionar uno de los motivos que se ofrecen para dar un reporte breve y claro.


Pulsamos sobre "Enviar" para enviar la evaluación y continuamos hasta que acabe la ronda, donde podemos volver a jugar otra ronda, o salir al menú del grupo.



Evaluier die Definition!

Auf Englisch sagt man "supermarket". Und auf Deutsch? **Supermarkt**

* Sind die Substantive groß geschrieben? 


☐ Ja ☐ Nein

 Report

 Senden  Speichern

Ver estadísticas

En el menú del grupo podemos ver nuestro progreso en ese grupo pulsando sobre "Estadísticas".

Juan Miguel Ruiz Ladrón	
Gespielte Definitionen	36
Erfolge	26
Meist gespielte Kategorie	Essen
Kategorie	
Name	Definition
Essen	13
Wetter	1
Wochentage	1
Aussehen	1
Grammatik	10
Definitionen mit Fehlern	
 Zurück	

C.1.2. Manual de la web

Manual de usuario: Web

En éste manual explicaremos:

- Registro y Login
- Crear grupo y modificarlo
- Añadir contenido al grupo y gestionarlo
- Validar alumnos y gestionarlos
- Generar estadísticas

Registro

En la ventana principal de nuestro entorno web nos aparecerá la ventana para registrarnos y, a la derecha, un pequeño botón blanco para ir a la vista de registro.

Clicquemos sobre "Register" y nos debe aparecer el siguiente formulario:

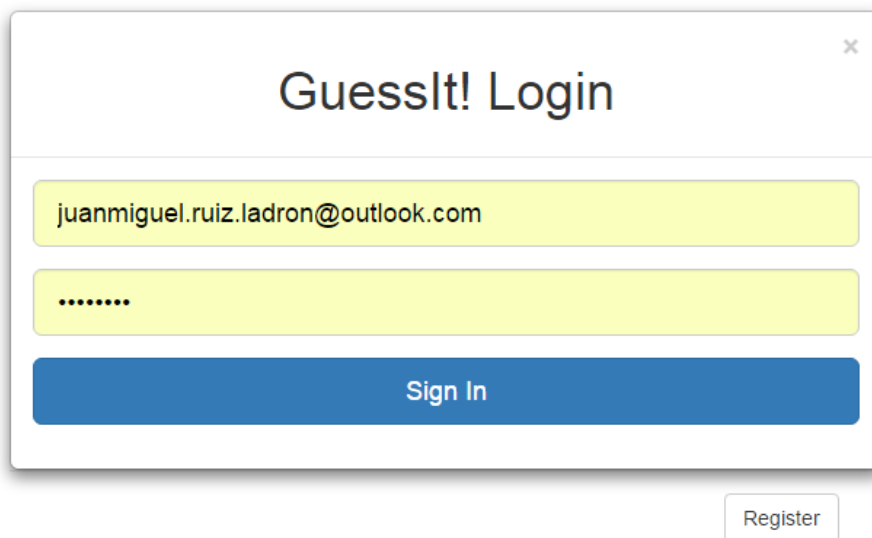
GuessIt! Portal Registration Form

Name:	<input type="text" value="Enter name"/>
Last Name:	<input type="text" value="Enter lastname"/>
Email:	<input type="text" value="Enter email"/>
Password:	<input type="password" value="Enter password"/>
Institution:	<input type="text" value="Enter studies center"/>
	<input type="submit" value="Submit"/>

Una vez relleno, pulsamos sobre "Submit" y esperamos a que el administrador del sistema nos de acceso.

Login

Una vez tenemos permiso para acceder a la plataforma, vamos de nuevo a la ventana de Login e introducimos nuestra credenciales de acceso.

A login dialog box titled "GuessIt! Login" with a close button (X) in the top right corner. It contains two yellow input fields: the first contains the email "juanmiguel.ruiz.ladron@outlook.com" and the second contains masked characters ".....". Below the fields is a blue "Sign In" button. At the bottom right of the dialog is a "Register" button.

GuessIt! Login

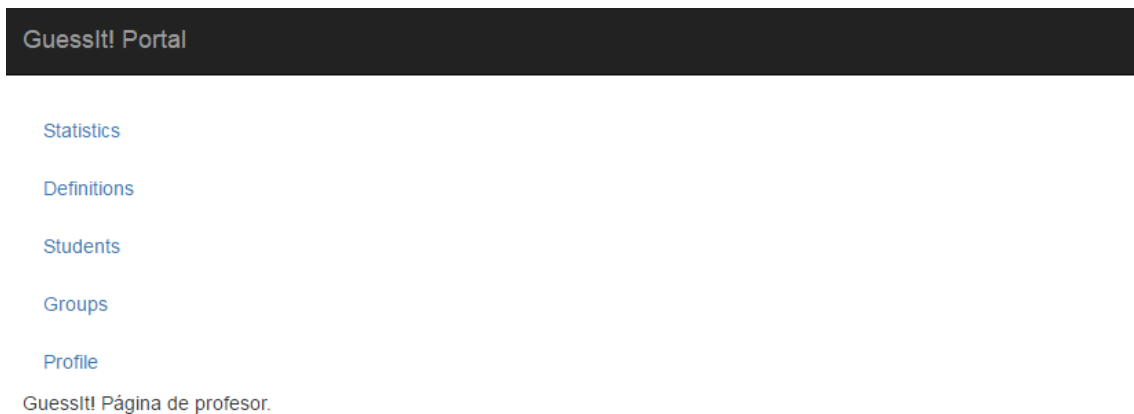
juanmiguel.ruiz.ladron@outlook.com

.....

Sign In

Register

Y entraremos en la ventana principal del entorno web:



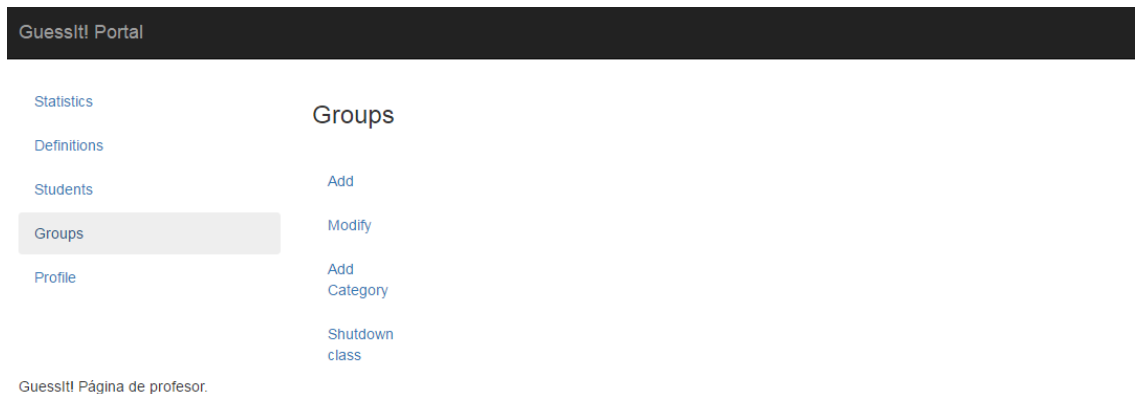
Grupos

El sistema se gestiona mediante el uso de grupos. Un grupo puede ser, o bien un aula, o una comunidad, o un grupo para un examen, cualquier cosa. Es la mecánica básica del sistema, todo se gestiona mediante grupos. Los alumnos piden acceso a los grupos y el docente los valida, así se evita que usuarios no permitidos manipulen la información del grupo.

Los pasos a seguir para crear un grupo son:

Crear grupo

Si pulsamos sobre "Groups" se nos despliega un submenú con varias opciones, pulsamos sobre "Add" para crear un nuevo grupo.



Nos aparece un formulario con:

- El nombre del grupo que queramos poner. Es público y se verá desde el cliente.
- El idioma del grupo. Los idiomas son creados por el administrador del sistema.
- Las categorías de palabras del grupo. Es muy importante no equivocarse en el nombre de las categorías, porque luego, si las definiciones son importadas desde CSV, éstas se conectan por la categoría. Las categorías se escriben separadas sólo por comas.

The screenshot shows the 'GuessIt! Portal' header. On the left is the same sidebar as before. On the right, under the 'Groups' heading, is a form with the following fields: 'Group name' (text input), 'Spanish' (dropdown menu), and 'Type categories separated with comma.' (text input). Below these fields is a 'Save' button. The sub-menu links 'Add', 'Modify', 'Add Category', and 'Shutdown class' are still visible. At the bottom left, it says 'GuessIt! Página de profesor.'

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Groups

[Add](#)
[Modify](#)
[Add Category](#)
[Shutdown class](#)

GuessIt! Página de profesor.

También se encuentra disponible un formulario de modificación de la información del grupo, por si nos hemos equivocado en algo.

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Groups

[Add](#)
[Modify](#)
[Add Category](#)
[Shutdown class](#)

Select Group:

Group's name:

Categories:

GuessIt! Página de profesor.

Se pueden tener todos los grupos que se quieran tener en la plataforma.

Definiciones

Existen varias operaciones para gestionar las definiciones de nuestros grupos, tales como: añadir, importar y modificar. También es posible ver las aportaciones y la ventana de corrección de reportes.

Añadir definiciones

Al clicar sobre la opción "Add" se nos despliega el siguiente formulario:

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Definitions

[Add](#)
[Import](#)
[Modify](#)
[Student's definitions](#)
[Report review](#)

Select Group:

PruebaMemoria ▾
Go!

Level:

Enter level

Word:

Enter word

Article:

Enter article

Definition:

Enter definition

Hint:

Enter hint

Category:

Casa ▾

Submit

GuessIt! Página de profesor.

Que rellenaremos para que nuestra definición quede así:

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Definitions

[Add](#)
[Import](#)
[Modify](#)
[Student's definitions](#)
[Report review](#)

Select Group:

PruebaMemoria ▾
Go!

Level:

1

Word:

perro

Article:

Enter article

Definition:

El perro es el mejor amigo del hombre.

Hint:

Animal canino cuadrúpedo.

Category:

Casa ▾

Submit

GuessIt! Página de profesor.

También podemos importar definiciones desde un fichero CSV, cuya estructura se encuentra al final de éste documento. Se recomienda el uso de Google Sheets para usar la plantilla y generar el CSV necesario para importar definiciones.

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Definitions

[Add](#)
[Import](#)
[Modify](#)
[Student's definitions](#)
[Report review](#)

Select Group:

PruebaMemoria ▾

Go!

Import csv:

Seleccionar archivo

 Ningún archivo seleccionado

Submit

GuessIt! Página de profesor.

También es posible modificar definiciones desde la opción "Modify", que nos mostrará un pequeño buscador de definiciones por categoría y nivel.

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Definitions

[Add](#)
[Import](#)
[Modify](#)
[Student's definitions](#)
[Report review](#)

Select Group:

PruebaMemoria ▾

Go!

Select the level:

Level 1 ▾

Select category:

Casa ▾

Go!

GuessIt! Página de profesor.

GuessIt! Portal

[Statistics](#)
[Definitions](#)
[Students](#)
[Groups](#)
[Profile](#)

Definitions

[Add](#)
[Import](#)
[Modify](#)
[Student's definitions](#)
[Report review](#)

Select Group:

PruebaMemoria ▾

Go!

perro

Article:
Definition: El perro es el mejor amigo del hombre.
Hint: Animal canino cuadrúpedo.

Modify

GuessIt! Página de profesor.

Usuarios / Estudiantes

Desde el menú "Students" tenemos todas las herramientas para gestionar a los estudiantes pertenecientes a cualquier grupo que hayamos creado. Podemos desde añadirlos desde cero, modificar sus datos en caso de errores y tenemos la opción de validarlos, si han solicitado unirse desde el cliente Android o el cliente de PC.

GuessIt! Portal

Statistics

Definitions

Students

Groups

Profile

Students

Add

Modify

Validate

Name:

Last Name:

Username:

Email:

Password:

Enter name

Enter lastname

Enter username

Enter email

Enter password

Submit

GuessIt! Página de profesor.

GuessIt! Portal

Statistics

Definitions

Students

Groups

Profile

Students

Add

Modify

Validate

Select Group:

PruebaMemoria

Go!

Juan Miguel Ruiz Ladrón

e-mail: juanmiguel.ruiz.ladron@gmail.com

joined: 2016-03-14 01:10:53

validate: ☐

Validate

GuessIt! Página de profesor.

GuessIt! Portal

Statistics

Definitions

Students

Groups

Profile

Students

Add

Modify

Validate

Select Group:

PruebaMemoria

Go!

Ruiz Ladrón, Juan Miguel

Email: juanmiguel.ruiz.ladron@gmail.com

Joined: 2016-03-14 01:10:53

Username: onemoretime

Modify

GuessIt! Página de profesor.

Generador de estadísticas

Por último, lo más importante, el generador de estadísticas.

Al pulsar sobre "Statistics" se nos abre el siguiente formulario:

GuessIt! Portal

Statistics

Definitions

Students

Groups

Profile

Select Group: PruebaMemoria

Informe por alumnos

Alumnos

Ruiz Ladrón, Juan Miguel

Informe

Número de definiciones jugadas

Tipo de informe

☐ Gráfica

☐ Tablas

☐ Fichero CSV

Informe por definiciones

Rango del informe

Inicio del rango YYYY-MM-DD

Fin del rango YYYY-MM-DD

Niveles

Nivel 1

Nivel 2

Nivel 3

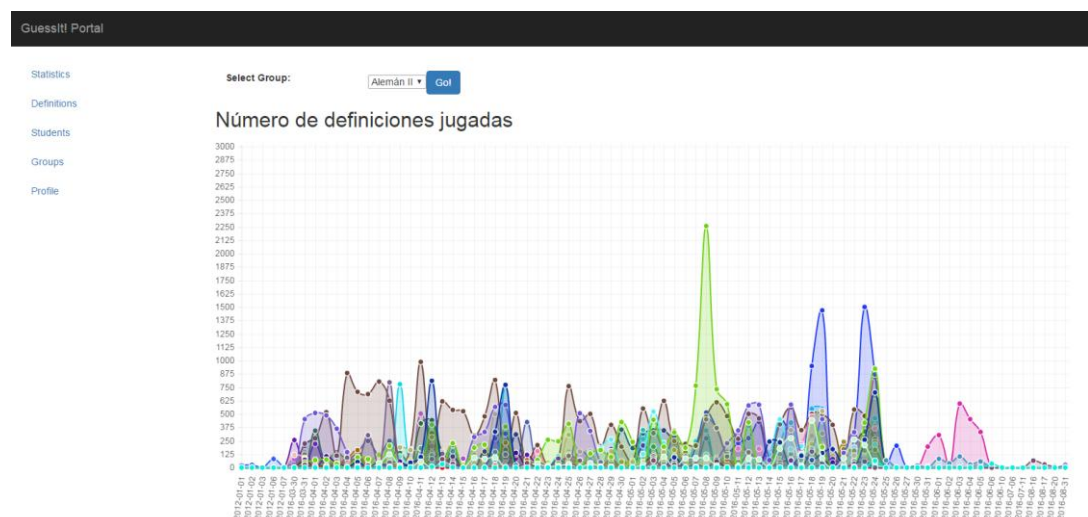
Nivel 4

Generar

GuessIt! Página de profesor.

Escogemos cualquier cantidad de estudiantes de nuestro grupo, seleccionamos un rango de fechas, en el formato que viene indicado (si lo dejas en blanco coge todo el tiempo de juego que lleva el grupo), el tipo de datos a generar, la forma de visualizarlos u obtenerlos y los niveles de las definiciones que deseemos comprobar.

Pulsamos sobre "Generar" y el resultado, una vez hayan jugado bastante, se asemejará a éste:



Es una gráfica interactiva, elaborada en Chart.js que nos muestra los datos que hemos seleccionado.

Anexo

Estructura del fichero Google Sheet

A continuación se muestra la estructura que deben llevar los datos para ser importados a través de CSV:

	A	B	C	D	E	F
1	Artikel	Wort	Niveau	Thema	Definition	Hilfe
2	das	Abendessen	4	Essen-Trinken	Mittags machst du dein Mittagessen und abends? Dein Abendessen.	Das Wort ist ein Substantiv und beginnt mit "A"
3	der	Alkohol	1	Essen-Trinken	Rum, Bier und Whisky sind Alkohol. Das Wort beginnt mit "A".	Es gibt Bier mit ... und ohne ...
4	die	Ananas	1	Essen-Trinken	Bob Esponjas Haus ist eine Ananas.	Wie sagst du "piña" auf Deutsch?
5	der	Apfel	3	Essen-Trinken	Auf Englisch sagst du "apple". Und auf Deutsch? Apfel.	Ein Wort mit "A".
6	der	Apfel	4	Essen-Trinken	Eine Frucht mit "A". Sie ist rot, gelb oder grün: Ein Apfel.	Auf Englisch "apple".
7	die PL.	Apfelsinen	3	Essen-Trinken	Ein anderes Wort für Apfelsinen ist Orangen.	Diese Frucht ist orange und beginnt mit "A".
8	das	Baguette	1	Essen-Trinken	Ein Brot aus Frankreich: Baguette. Das Wort beginnt mit "B".	Ein anderes Wort für Brot.
9	das	Bier	1	Essen-Trinken	Ich trinke ein Bier von Heineken, Mahou oder Cruzcampo.	Ein Getränk mit "B".
10	das	Brot	3	Essen-Trinken	Ein anderes Wort für Brot ist Baguette. Das Wort beginnt auch mit "B".	Auf Spanisch sagst du "bollo".
11	das	Brötchen	4	Essen-Trinken	Ein kleines Brot ist ein Brötchen. Der erste Buchstabe ist auch "B".	Auf Spanisch sagt man "una pieza pequeña".
12	die	Butter	4	Essen-Trinken	Auf Englisch sagt man "butter", auf Französisch "beurre" und auf Deutsch? Butter.	Es gibt Margarine und ...
13	der	Durst	4	Essen-Trinken	Wenn du Durst hast, dann trinkst du Wasser usw. Das Wort beginnt mit "D".	Wie sagst du "sed" auf Deutsch?
14	der	Kuchen	3	Essen-Trinken	Ich backe einen Kuchen. Der erste Buchstabe ist "K".	Ein anderes Wort für Torte.
15	die	Marmelade		Essen-Trinken	Ein Wort mit "M". Viele Leute essen morgens Brot mit Butter und Marmelade.	Auf Englisch "marmelade".
16	die	Banane	1	Essen-Trinken	Was ist gelb und kommt von den Kanarischen Inseln? Die Banane.	Eine Frucht mit "B".

Aunque esté escrito en Alemán, la estructura se compone de 6 columnas:

- Artículo: en caso de que la palabra disponga de artículo (que en Alemán casi todas tienen).
- Palabra: la palabra a adivinar.
- Nivel: nivel de dificultad de la palabra
- Categoría: Importante escribirla bien para evitar errores en la subida.
- Definición: La frase o pregunta que **DEBE** contener la palabra a adivinar, el cliente se encarga de esconderla luego. Aunque sea una pregunta, añadir la palabra al final de la pregunta.
- Pista: Una pista que ayude a adivinar la palabra.

Bibliografía

- [1] Preguntas y respuestas:
<http://stackoverflow.com/>
- [2] Documentación de LibGDX:
<https://libgdx.badlogicgames.com/documentation.html>
- [3] Documentación de CodeIgniter:
<https://www.codeigniter.com/docs>
- [4] Documentación oficial de PHP:
<http://php.net/docs.php>
- [5] Informe virtual *Radar tecnológico 2016*:
<https://assets.thoughtworks.com/assets/technology-radar-apr-2016-es.pdf>
- [6] Libgdx Cross-platform Game Development Cookbook, de Alberto Cejas Sánchez y David Saltares Márquez:
<https://www.packtpub.com/game-development/libgdx-cross-platform-game-development-cookbook>
- [7] Tutoriales de LibGDX:
<http://www.gamefromscratch.com/page/LibGDX-Tutorial-series.aspx>
- [8] Manual web de HTML y SQL:
<http://www.w3schools.com/>
- [9] Documentación de Java:
<https://docs.oracle.com/javase/7/docs/api/>
- [10] Instalación del servidor:
<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu>

GNU Free Documentation License

GNU GENERAL PUBLIC LICENSE

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License.

If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and

control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the

Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.

SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <textyear> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses>

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>